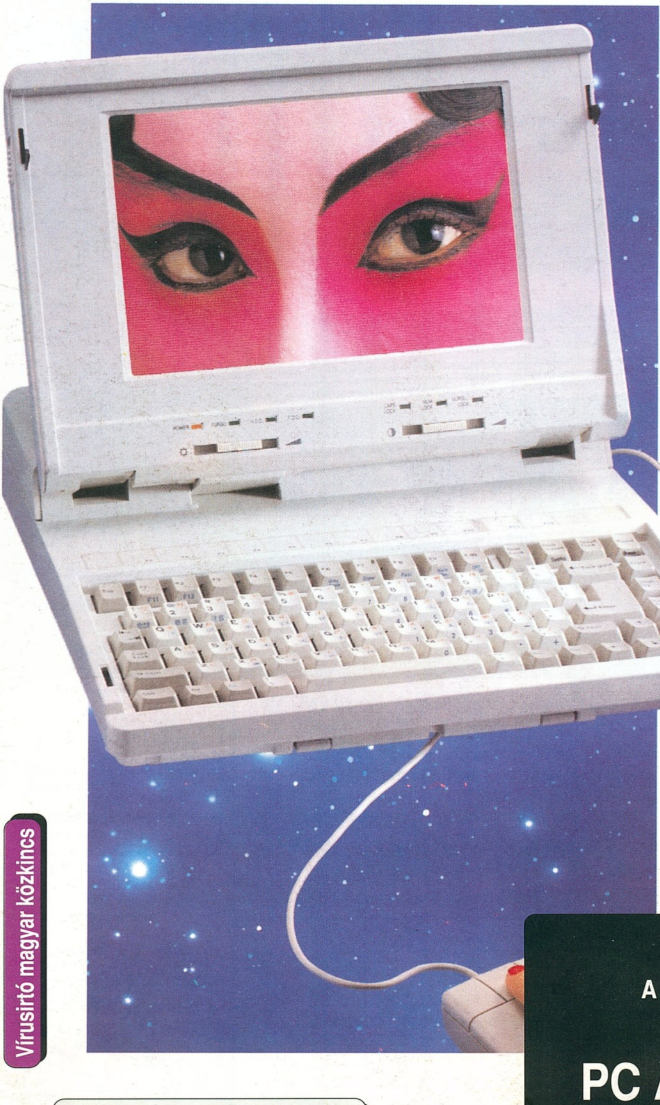
1992 / JANUÁR

ÁRA: 196 FT

ALAPLAP

MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL



Tmrttt állományok kiteeríiítveeee

A NyelvÉsz és a Lektor

Takarodó a rezidensnek

Dekoráljunk CAD-élyesebben!

A MÁGNESLEMEZEN:

Beszállókártya
Egy csomagdemó
Az Alaplap eddigi számainak tartalma
Játék: Ugrabugra

Cluster Buster

Ventura 4.0 — Színek az Ablakban

A sorsolás nyertesei

Vírusirtó magyar közkincs

A HÓNAP TÉMÁJA:

PC A ZSEBBEN

Calamus(zi) macska egeret fog

Ha a megbízhatóság a döntő...



VIGYÁZAT! Jól bevezetett és hírnévnek örvendő márkanevünkkel kétes minőségű, hasonló hangzású nevek élnek vissza!

A MITAC 17 éves információipari hátterével a technológia egyik távol-keleti vezetője. Igen szigorú minőségbiztosító rendszerének és hatalmas kutató-fejlesztő beruházásainak eredményeképpen termékei a világ 65 országában váltak a korszerűség és a megbízhatóság szinonimájává.

A megbízható gyártó termékei csak megbízható forgalmazó tevékenysége nyomán képesek a felhasználó javát szolgálni.

Ezért esett a MITAC választása hazánkban az INTERAG-ra.

Forgalmazó: INTERAG

INTERAG INFORMATIKA
Budapest 1136 Pannónia u. 11.
Tel./fax.: 132-9375 Molnár Péter

People Committed To InfoTech

MITAC

INFORMÁCIÓKÉRÉS: 01 ▶

ALAPLAP

Mikroszámítógép magazin mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Munkatárs:
Sziebig Andrea

A mágneslemez melléklet és a Közkincs szerkesztője:
Verebély Pálné

A Lemezkalauz szerkesztője:
Nagy Gábor

A szerkesztőbizottság tagjai:
Barna László, Boros György, Broczkó Péter, Brüll Károly, Farkas Ernő, Herczeg József, Kassay Árpád, Kónya László, Kovács P. Attila, Pintér Gábor, Vargha Dénes, Vékony Tamás Villányi László, Zoltai Péter

Szerkesztőség, kiadó és hirdetésszervezés:

VIII., Reguly Antal u. 8.
Budapest 1441
Telefon és fax: 133-1839

Felelős kiadó:
Sebestyén Ilona
ügyvezető igazgató

Cédrus Kiadó Kft.

Nyomdai előkészítés:
Tipoprint Kft., Budapest
Nyomtatás:
Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta. Előfizethető a hírlapkézbesítő postahivataloknál és a Posta Hírlapelőfizetési és Lapellátási Irodájánál (XIII., Lehel u. 10/a, Budapest 1900), vagy átutalással a 215-96162 pénzforgalmi számra.

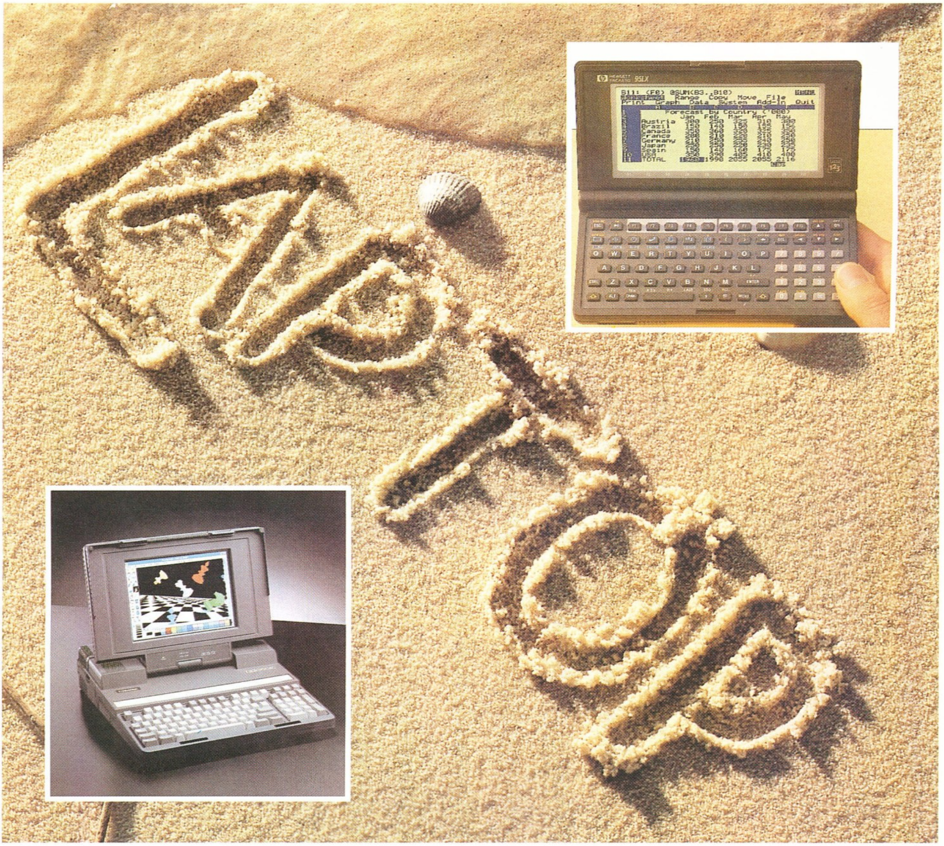
Példányonkénti ár: 196 Ft
Évi előfizetési díj: 2 352 Ft
PC Turbo Klub-tagoknak: 2 112 Ft
(Tagfelvétel a szerkesztőségben)

Külföldre terjeszti a Kultúra, Pf. 149, Budapest 1389

HU ISSN 0865-9788

A HÓNAP TÉMÁJA: PC A ZSEBBEN

Címlapképünk a tajpeji számítástechnikai vásár reklámgrafikája



A csecse becse

Bárki, aki zseb- vagy hordozható számítógépet, vezeték nélküli telefont, kalkulátort, hordozható CD-lemezjátszót, sétamagnót, videokamerát, digitális órát használ, a gyakorlatban érzékeli a hordozható berendezések előnyeit. Nem kell tápegység, könnyű velük mozogni, nincs hatalmas vezetéktömeg.

1981-ben Adam Osborne nem csupán egy újfajta számítógéppel jelentkezett, hanem egy új tervezési filozófiát is átültetett a gyakorlatba: ez volt a hordozható számítógép, amely egy nagyobb táska formáját utánozta, de a hozzá adott hálózati csatlakozó jelezte, hogy csak külső áramforrásról működik.

Amikor pedig 1983-ban a Tandy megjelent a Model 100 típusú géppel, amely hálózatról és telepről is működött, a hordozható gépek családjából kivált a laptop gépek családja. Ha fontos volt a bárhol való használhatóság, akkor a kevesebbet tudó, de telepről működő laptop volt a megoldás. Ha csak a hordozhatóság, a mobilitás volt fontos, akkor a nagyobb teljesítményű, kompakt mechanikával felvértezett hordozható gép volt célszerű. Azóta viszont ez a két család folyamatosan közelít egymáshoz, ami elsősorban a jobb hatásfokú telepeknek és a kisebb fogyasztású rendszerelemeknek köszönhető.

Összeállításunk — a valódi csecsebecséktől a noteszgépekig — elsősorban a hordozható gépek ma is meglévő gyenge pontjaira, illetve a továbblépés érdekében tett erőfeszítésekre koncentrál. A perspektívák felvillantását a piaci megközelítésű írások és a szakirodalom forrásanyagai szolgálják.

A telefonnotesztől a palmtopig

Gonosz törpék fortélyai

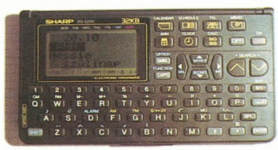
„A ma titkára nem kap fizetést, pedig éjjel-nappal dolgozik, és a kávéfőzésen kívül mindenhez ért" — reklámozta a Sharp az IQ-7000 Electronic Organizert. Vajon tényleg mindent tud? Vagy ez is csak üres reklámszöveg? Sokan legyintenek: komolytalan játékszerek. Vannak viszont, akik szinte már zoknit is csak akkor váltanak, ha erre a menedzserkalkulátor kifejezetten felszólítja őket. Az alábbiak alapján ki-ki döntse el maga, van-e rá szüksége.

A legtörpebb törp

Hajdanvolt barátnőm második gyermekét várja (fájdalom, nem tőlem), de szüleinek telefonszámára ma is emlékszem. Hiába hívnék azonban jó néhány számot, csak bosszúság forrása lenne, mert Ikszék elköltöztek és/vagy a szám megváltozott... A folyamatosan aktualizált címekkel-telefonszámokkal egymás után telnek meg a noteszek a papírgyárak nagy örömére, miközben a Föld naponta veszíti el erdő-tüdejét. A megoldáshoz már nem kell a meleg vizet feltalálni: segít a számítógép. De vajon érdemes-e azért a telefon mellett egy külön PC-t tartani, vagy akárcsak a Wordstarból kilépni és betölteni a dBase-t (pardon, Kisokost), hogy egy szem telefonszámot kikeressünk? Hamar visszaszoknánk a jól bevált papírra-ceruzára.

Szerencsére a hardvergyártók is a piacról élnek, s mára már bőséges a kínálat az elektronikus noteszből. Hogy valóban mindig kéznél lehessen, a legkisebbek hitelkártya-méretűek, s a nagyobbak sem nőnek túl egy szokványos zsebkalkulátoron (1. kép). Vagy ötvenszáz név—telefonszám párosítás tárolására képesek, de a nevet említve ne gondoljunk a spanyol nemesi cirkalmakra, ugyanis az





alfanumerikus mező nyolc karakterre korlátozott. Az érintő vagy legfeljebb gumiharang-érintő típusú „billentyűzet" nem is csábít nagyregények begépelésére. Az ABCD-elrendezés még hagyján, de az, hogy helytakarékosságból egy gombra három betű is jut, olyan, mint egy versenyfutás letolt nadrágban. Az O betű beírásához az MNO5 gombot éppen háromszor kell lenyomni. Különdíjat érdemel, aki magától rájön, hogy mondjuk a „Soós" névben az egymás melletti két o betűt hogyan lehet beírni. (Újbóli nyomkodásra ugyanis csak az MNO5-MNO5 karaktereket ismételgeti konokul ugyanazon a helyen.) Azért említsük meg egy előnyét is: nagyon olcsó! Egyes példányok már 5 DEM-ért is megkaphatók.

Pittyegő zsebtitkárok

Ismerik-e a szórakozott professzor esetét, akinek egy fontos előadása akadt, s hogy el ne felejtse, feladott magának egy levelet. Kár a sok szóért, ennyit írt bele: Előadás! — majd mikor megkapta, mérgesen tépte szét: ugyan, ki szórakozik velem? Hasonló érzés, amikor megszólal a karunkon a kvarcóra: a kollégák vicceltek, vagy most kell az igazgató úrhoz menni?

A zsebtitkárnak tehát rendelkeznie kell tetszőleges számú és idejű „ébresztés" programozásának lehetőségével, úgy, hogy tájékoztatást is adjon a riasztás okáról. Ezt a feladatot többféle módon, kisebb vagy nagyobb eleganciával lehet megoldani. Vegyünk a Sharp választékának két végéről egy-egy készüléket! Az EL-6190 (2. kép) a kínálat aljáról való. 8 kB memóriáján két program osztozik: a telefonkönyv és az ütemező (scheduler). Ez utóbbiban évekre előre megrendelhetjük az ébresztést, időben figyelmeztet a házassági évfordulóra vagy a randevúra a titkárnővel. A telefonkönyv érdekessége, hogy mindjárt három notesz kínál (TEL1, TEL2, TEL3). Az elsőben ismerősök, barátok nevét tartom, a másodikban hivatalokét, a harmadikban bankszámlaszámokat stb.

Attól nem kell tartanom, hogy valamilyen adatot illetéktelen felhasználhasson: bármi kijelölhető titkosnak, ekkor csak az előzetesen megadott jelszó beírása után tekinthető meg. Bár a név ezen a gépen is csak tizenkét karakter hosszú lehet, a pötyögtetés jóval kellemesebb a QWERTY-tasztatúrán. Azonos külméretekkel a funkciók sokszorosát nyújtja a ZQ-2250 (3.kép). „Kép-



ernyője" 12 oszlop x4 sor méretű, minden sor alfanumerikus, ezért a név nincs többé korlátozva, sőt cím is tárolható. A noteszokban a tetszőleges karaktersorozat megkereshető, mint egy jobb szövegszerkesztőben. A gép rendelkezik az IBM karakterkészlet javával, plusz találhatunk telefon, pohár, harang, vészkijárat és sok más hasonló hasznos szimbólumot.

Itt nemcsak a telefonkönyvet vágták szét, hanem az „ébresztéseket" is: külön tárolhatjuk az üzleti tárgyalásokat és a családi évfordulókat. A gép rendelkezik öröknaptár-funkcióval is, itt számításokat is végezhetünk, például: a mai dátum + 38 nap = ? A 32 k memória és a QWERTY-billentyűzet lehetővé teszi a gyors jegyzetek készítését minimumszövegszerkesztő funkciókkal.

Sic itur ad astra

Az előbbiek mind nagyon szépek, de mennyivel jobban tudná ezt egy PC az xxx programmal! — sóhajt fel a nyájas olvasó. Dehát az egy nagydarab, ormótlan szerkezet, nem vihetem magammal a zsebemben... Fellélegezhetünk, kollégák, itt vannak a palmtopok, azaz a tenyérben is elférő, videokazetta méretű MS DOS-kompatibilis számítógépek, mint a júniusi számban ismertetett Atari Portfolio vagy a Hewlett-Packard 95LX. A Portfolio ugyan ROM-ban tartalmaz szövegszerkesztőt, adatbázis- és táblázatkezelőt, de ha nem vagyunk elégedettek vele, betölthetjük a WordStart, a Lotust... Ennél már csak egy igazi titkárnő lehet jobb ... néhány tekintetben.

Zoltai Péter

Momenta Pentop

Jegyzeteljünk fényceruzával!

A számítógépes szakemberek egy részét meglepte az a mohó érdeklődés, amellyel a fényceruzával kezelhető számítógépeket tágabb szakmai körökben — sőt a laikusok is — fogadták. „Hát nem szép és jó ez így együtt? Billentyűzet, egér, monitor... mi kell még?" Ezekhez valóban nem kell már semmi! Helyettük kell a „palatábla" és az „íróvessző".

Ha a kézírásunk bonyolult, változékony és egyéniesített jelrendszerének számítógépes dekódolásán alapuló új technológia beérik, személyes használati tárgyaink sorába nagyon gyorsan bekerül a valódi elektronikus notesz, amely megőrzi számunkra a ceruzával írás kényelmes (és évezredes beidegződésű) szokását. Forradalmi újdonság startjának vagyunk tehát tanúi, amely messzemenően kihat majd a számítógépek valamennyi kategóriájára, és általa a táskagépek, de leginkább a noteszgépek teljesen új jelleget öltenek majd.

Az új technológia kidolgozását célzó versenyben az eddigi 3 cég (Go, Grid, Microsoft) mellé felzárkózott — és elsőként a piacra lépett — az ugyancsak amerikai Momenta, több szempontból az eddigi megoldásokétól eltérő filozófiával. Ezek közül legfeltűnőbb, hogy a Momenta nem mond le teljesen a billentyűzet használatáról, mert a kézírásfelismerő technológiát még nem tartja elég kiérleltnek minden funkció átvételére. Ennek megfelelően a fényceruza sok esetben tulajdonképpen az egér szerepét tölti be. Magyarul a fényceruza szót tartom legtalálóbbnak, de a toll (= pen) vagy az elektronikus íróvessző (= electronic stylus) megnevezés is használatos.

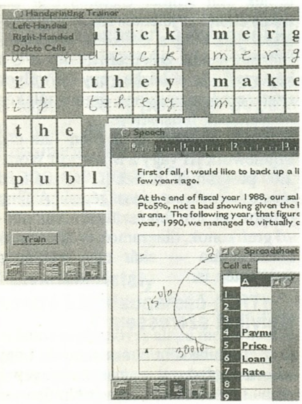
A Momenta gép lelke egy 20 MHz-es, 386SX processzor, amit 4 MB RAM, 1 MB ROM és 40 megabájtos merevlemez támogat. A többfunkciós képernyő 16 tónusú, 640 x 480 pixeles VGA. A Momenta grafikus felhasználói felülete valahol félúton van a DOS-sal nem kompatibilis PenPoint (Go) és a DOS-alapú Pen Windows (Microsoft) között. A Momenta rendszeren ugyanis futtathatjuk a DOS programokat, de nem a fényceruzát közvetlenül használva, hanem közbeiktatva Smalltalkra épülő, saját tárgyorientált grafikus felhasználói felületét.

Abban minden eddigi fényceruzás rendszer megegyezik, hogy a kézírásfelismerő programban tárolt (általában mintegy 300), eltérő írásmódot tartalmazó mintát tovább bővíthetjük, tehát a gépet „megtaníthatjuk" saját (olvashatatlan) írásunk elolvasására is. Arról azonban csak hosszas használat során lehetne elegendő tapasztalatot szerezni, hogy a programok jelfelismerő képességének határa jelenleg hol húzódik. (Pénztárcánk határai azonban ismeretesek, és a Momenta ára Németországban kereken 10 ezer márka.) Jegyzetelésünket azután „fakszimileként" és ASCII



szövegállományként egyaránt tárolhatjuk.

A fényceruzás számítógépek jövőjéről nyilvánvalóan nem lehet a még sok gyermekbetegséget magukon hordozó „első fecskék" alapján ítéletet alkotni. Másrészt az új találmányok megjelenésekor szokásos túlzó jóslatokat is érdemes elkerülni. A fényceruza egyeduralma a noteszgépek kategóriájában valóban elképzelhető. A táskagépeknél ál-



talános alkalmazására — alternatív kezelési módként — szintén nagy valószínűséggel lehet számítani. Arra azonban nem, hogy a munkahelyi és otthoni személyi számítógépek mellől eltűnik a billentyűzet és az egér, s helyette az ember és a gép között minden interaktív művelet egyetlen képernyőn s egyetlen fényceruzán keresztül zajlik majd le. A billentyűzet ugyanis a szövegbevitelnek a kézírásnál sokkal gyorsabb, termelékenyebb módja, amiről nem lehet lemondani, még akkor sem, ha a kézírás felismerésének pontossága egyszer valóban eléri a 100%-ot. Ez utóbbitól azonban elég messze vagyunk, s a jelenlegi 90-95% feljebb tornászásához a jelfelismerésen kívül igen bonyolult és nagy számításigényű grafológiai és szemantikai elemzési műveleteket is integrálni kellene a programba, amihez ma még kellő elméleti és módszertani támpontok sincsenek.

Bárhogy is alakul a kézírás felismerésén alapuló, fényceruzával kezelhető számítógépek térhódítása, az bizonyos, hogy ismét kreatív gondolkodásra késztető technikai lehetőség került a számítógépesek asztalára — nehogy elkényelmesedjenek a sok rutinmunkában.

Faklen Pál

Aki fut, aki nem...

A „felnőtt" PC-k világában is az egyik legmisztikusabb téma az összeférhetőség kérdése. Eltérő hardver, eltérő BIOS, DOS-verzió és több, hasonlóan nehezen felderíthető különbség keseríti a felhasználók életét. Eleve gyanakvással fogadja az ember az állítást az olyan zsebtörpe kompatibilitására vonatkozóan, amely hardverszempontból látványosan különbözik az őseitől, mint például az Atari Portfolio. Elindul ezen egyáltalán valami?

Ami nem megy...

Nem akarok olyan olcsó poénokat elsütni, hogy például nem indul el a VGA-kártyát igénylő xxx játék. Ezt „a vak is látja". Tapasztalatom szerint nem működnek a „hangos" programok. Lehet az bár a legegyszerűbb CEDSIG.COM, a sokak által elátkozott tutulás a lemezmelléklet elején; lehet az bármilyen trükkös háttérzene: „eltörött a hegedűm, nem akar szólani". Itt semmi konkrétumot nem tudok, hogy miért. Nem működnek azok a programok, illetve azok a funkciók sem, amelyek az Alt + valamilyen billentyű kombinációjára indulnak el: ilyen az ASC.COM — pedig milyen jó lenne a lemezmellékleten megjelent memóriarezidens ASCII-táblázat. Ennek oka a hiányzó ANSI.SYS lehet: sajnos pótlása akárhonnan leszedett ANSI.SYS-szel nem járható út.

...és ami elindul

Nagyon örültem a DOSEDIT.COM működésének, ami nem Norton Commander ugyan, de a DOS-parancsok beviteléhez hathatós segítséget nyújt. A hossza, azaz inkább a rövidsége pedig (1920 bájt) mintha egyenesen a szűkös memóriájú Atarinak predesztinálná. Az Alt billentyűvel elérhető funkciók sajnos nem működnek. A játékok kipróbálásakor meglepetést szerzett a LABY.COM, amely elindul ugyan, sőt végig is játszható, a Portfolio kedves ajándéka viszont, hogy nem csak a „pofa" környezetét láthatjuk, hanem a teljes labirintust — illetve annak a kijelzőn lévő szeletét. Ez feltehetően az eltérő képfrissítési mechanizmus következménye.

XDEL.COM — szinte már nem is parancs, hanem utility: a parancssorban megadott fájl(oka)t felsorolja, majd beleegyezésünk esetén törli. A jokerek természetesen használhatóak: az XDEL *.* tavaszi nagytakarítást tart, az XDEL *.TXT csak a szövegeléseinket törli le. Vigyázat! A Read-Only flag sem akadály, az így kijelölt fájlokat is törli minden megjegyzés nélkül!

ATTR.COM — az előbb említett flageket lehet beállítani (hidden, read-only). Bár erről az Atari gépkönyve mélyen hallgat, azért működik!

WAIT.COM — a másodpercben megadott idő eltelte vagy billentyűlenyomás után engedi tovább a batchfájlok futását.

ASK.COM — itt a felsorolt billentyűk egyikének lenyomásáig tartja fel a batchprogramot; hogy melyik billentyűt püföltük, az az ERRORLEVEL változóból derül ki.

INPUT.COM — ez a ravasz kis program megismétli azt, amit beírtunk neki.

Botrány hardverügyben

A PC-vel való kapcsolattartásnak három lehetséges útja-módja van. A Portfolio RAM-kártyái a PC-n is használhatók, ha van megfelelő Card Drive egységünk. Kapható soros és párhuzamos interfész is. A soros mellé semmilyen programot nem mellékelnek, hiszen lehet, hogy nem is PC-vel, hanem Amigával vagy Atari ST-vel fogjuk összekötni törpénket. Mindenki szerezze be, ahogy tudja.

Természetesen kutasson az illető a Portfolión futó terminálprogram után is, merthogy ahhoz is kell. A Markt & Technik kiadó „Portfolio Praxisbuch"-jának (ára 2790 Ft!) lemezmellékletén található egy XTerm nevű program, amely azt állítja magáról, hogy tudja az Xmodem protokollt. Nos, lehet, hogy tud valamit, de az biztos nem az Xmodem — a Telix ugyanis semmit sem „hall" belőle. Az a helyzet, hogy az interfész és a könyv együttesen tízezer forintos költsége kidobott pénz: a PC-vel kapcsolatban egyik sem jó semmire! Szerencsére a párhuzamos csatoló, ha lassan is, de tökéletesen működik, s ezzel még nyomtatóhoz is kapcsolhatjuk a Portfoliót. Az igazi csattanó most következik: ez utóbbi, programmal együtt, feleannyiba kerül, mint az első csupaszon!

Z. P.

Szoftverszemüvegben

A displayt látni és meghalni

Az ember—számítógép kapcsolat mind ez idáig legfontosabb eszköze a képernyő, amelynek szerepe különösen a PC-k megjelenésével vált hangsúlyossá. Azonban mint annyi minden, ami a számítástechnikával kapcsolatos, a monitor sem kerülhette el végzetét: a miniatürizálást. De vajon kedvenc noteszgépünk képernyőjén hogyan festenek kedvenc szoftvereink?

Kezdetben vala a menü...

Az utóbbi néhány évben megjelent tisztességes szoftverek mindegyike rendelkezik valamilyen menürendszerrel. Ezek lehetnek kijelölő (highlight), legördülő, ablakos stb. típusúak. Asztali PC-n egy jól szervezett és esztétikusan kivitelezett menürendszer szemet gyönyörködtető látvány. Sok notebook és laptop gépen viszont szem(orvos)erőltető és fogcsikorgattató — tisztelet a VGA-s vagy a még ritka (és drága) színes képernyős kivételűeknek.

Bár már egyre több szoftvert felkészítenek a noteszgépekkel való találkozásra (ilyen például a jó öreg Norton Commander konfigurációs menüjének „laptop" pontja), azért még mindig gyakran találkozhatunk furcsán árnyékolt vagy a láthatóságot messzemenően elkerülő színeket alkalmazó menürendszerekkel. Kiváltképp érdekes effektusokat produkálnak a menüdobozok megjelenései a különböző háttérvilágításos LCD monitorokon. Sok esetben légies szellemkép jelenik meg a doboz alatt a képernyő teljes szélességében. Ezen olykor segíthetünk a fényerő beállításával. Persze a szoftvereknek is igazuk van valahol. Gondolok például a programozók számára készült fejlesztői környezetekre, melyekben igazán nem előírás a noteszgépek képernyőspecialitásainak figyelembe vétele, hiszen programot fejleszteni notebook gépen úgyis felér egyfajta öngyilkossággal.

Egyesek és mások...

Vannak persze igazi, profi notebook-szoftverek is, melyek kimondottan a notebook alkalmazási körének megfelelő feladatok ellátására íródtak. Ilyen például a DeskMate felhasználói felület, amit a Tandy különböző notebookjaihoz már alapszoftverként szállít. A DeskMate saját oktatóprogrammal rendelkezik, melynek segítségével bárki elsajátíthatja a szövegszerkesztő, a menedzsernaptár és a többi beépített alkalmazás használatát. A szoftver automatikusan képes konfigurálni a környezetben üzemelő monitort, de egy hasznos kis segédprogram segítségével kézzel is beállítható a megfelelő paraméter. Ha ismeretlen típusú monitorral kerül szembe, akkor sem küldözget kétértelmű hibaüzeneteket, hanem javasolja a képernyőhöz szállított meghajtóprogram alkalmazását. Futása közben bármilyen típusú monitor esetén esztétikus, legfőképpen pedig jól látható képernyőket produkál.

Általában a szövegszerkesztők gond nélkül futtathatók (és láthatók) noteszgépeken, mivel ez a tevékenység szorosan hozzátartozik az ilyen gépek alkalmazási köréhez. Az elterjedtebbek közül a Microsoft Word 5.0 jól futott a Tandy 1500 HD CGA jellegű és a Walkom NP-902 VGA monitorán, méghozzá bármiféle képernyőmeghajtó alkalmazása nélkül. (A Print menüpont PreView opciójának használatát mellőztem.) De ugyanígy használható a WordStar, a Kedit és sok más szövegszerkesztő is.

Nem mondható el ugyanez a különféle grafikus vagy egyáltalában grafikát használó programokról ilyen egyértelműen. A színes monitorral rendelkező gépek kivételével a 4–16 szín árnyalatokkal való helyettesítése már jelentősen próbára teszi a felhasználó szemét; nem beszélve arról, hogy a monitor beállításának a szemhez képest tökéletesnek kell lennie. Ugyanis már pár fokos eltérés is jelentős láthatóságvesztéssel jár. Még a 640x480 képpontos felbontású, 32 szürkeárnyalatos VGA képernyőn is nehézkes egy bármilyen tiszta, eredetileg színes, digitalizált kép részleteinek felismerése.

Ennek ellenére az Asthon-Tate cég Framework III vagy a Microsoft Works 2.0 programja remekül használható szinte bármilyen noteszgépen. A fenti programok írói felkészültek a különböző grafikonok, diagramok és táblázatok ábrázolására — akár egy repülőgép fedélzetén is.

Melyiket szeressem ?

Összefoglalásként elmondhatjuk, hogy sem a szoftverek, sem a noteszgépek nem hibáztathatók egyértelműen az esetleges összeférhetetlenségért. A noteszgép-gyártók tobzódnak a különféle képernyőkben, a szoftverírók pedig igyekeznek lépést tartani velük. Az upgrade verziókban mindig felbukkannak újabb képernyőmeghajtók, melyek igyekeznek megfelelni a notebook-konjunktúra kihívásainak, de érdemes háromszor is meggondolni egy-egy program ilyen jellegű, huzamos alkalmazását. Tudomásul kell vennünk, hogy a noteszgépek is csak meghatározott — bár egyre bővülő — feladatkörben alkalmazhatók teljes sikerrel.

Tass Csaba

Ábrázolástechnikák három betűvel

LCD, PDP, TFT, STB

A hordozható számítógépek megjelenésével a katódsugárcsöves megjelenítők elvesztették egyeduralmukat a megjelenítők piacán. Helyükre lassan a folyadékkristályos és a gázplazmás megjelenítők lépnek. Az LCD monitorok elsősorban a notebook, a laptop és a palmtop gépeknél, míg a PDP megjelenítők az ipari alkalmazások területén gyakoriak. De vajon mit takarnak ezek a rövidítések? Merre tartanak a lapos megjelenítők fejlesztői, gyártói? A választ a Display Device című szakkiadvány alapján ismertetjük.

Folyadékkristály kontra katódsugárcső (CRT)

A katódsugárcsöves alapú megjelenítők mint a monitorok első formái ma is és a közeljövőben is a legelterjedtebb információközvetítő eszköznek számítanak. Hátrányuk azonban az alacsony nyomású gázt tartalmazó, masszív, nehéz üvegtartály, ami jelentősen gátolja a két végletbeli — a hordozható és az óriás megjelenítők — méretben történő alkalmazását. A többirányú fejlesztések közül széles alkalmazási lehetőségei miatt a folyadékkristály technológia látszik hivatottnak átvenni a katódsugárcsöves helyét. A folyadékkristály három fő jellemzője, hogy optikai és elektroanizotróp, valamint folyékony. Az anizotrópia a kristályos anyagok azon jelensége, hogy egyes tulajdonságaik különböző irányokban különbözőek. Amikor a folyadékkristály elektromos mezőbe kerül, megváltozik molekuláinak elhelyezkedési iránya az elektroanizotróp tulajdonsága és a folyékonyság miatt. Ez az elhelyezkedésváltozás látható optikai változásként jelenik meg. A hatást, amelynél elektromos töltés hatására valamilyen optikai tulajdonság változik meg, elektrooptikai hatásnak nevezzük. Ezen a hatáson alapul az LCD-k működése.

A főbb megjelenítési módok

Többféle megjelenítési mód létezik, melyek az optikai anizotrópián alapulnak. Az első kifejlesztett megjelenítési mód az ún. DS (dynamic scattering = dinamikus szóródás), amit az első elektronikus számológépeknél használtak. Ha a folyadékkristályt két egymástól 15-80 mikrométer távolságra lévő átetsző lap közé tesszük, a molekulák a folyadékban a felületek közti feszültség miatt fix alakzatban helyezkednek el. Ilyenkor a folyadékkristály átlátszó. Amikor azonban elektromos áramot vezetünk a folyadékkristályba, a molekuláris hőáramlás miatt ez az alakzat felbomlik, és a molekulák „össze-vissza" helyezkednek el. Emiatt változik meg a folyadékkristály fénytörése. A két állapot közötti átkapcsolással válik lehetővé a megjelenítés. A DS mód hátránya a gyenge kontraszt, a korlátozott használhatóság a megvilágítás miatt és a nagy áramfelvétel.

A TN (twisted neumatic) módszernél a folyadékkristály már csak egymástól néhány mikrométerre elhelyezett felület között helyezkedik el, így a kristály molekulái 90 fokkal elmozdulnak tengelyük mentén. A két lapot pedig, melyek között a folyadékkristály elhelyezkedik, szendvicsszerűen két másik polarizáló lap fogja közre. Így az érkező fényt a lapok polarizálják, mielőtt a folyadékkristály réteget elérné. Amikor a fény eléri a folyadékkristály réteget, 90 fokos szöget zár be annak molekuláival. A fény ezután keresztülhalad a rá merőleges következő polarizáló lapon. Ha ekkor a folyadékkristály elektromos mezőbe kerül, megváltozik molekuláinak szöge az előző állapothoz képest. A fény csak az elektromos mező megléte esetén tud a két polarizáló lap között áthaladni. A TN módú megjelenítés a legnépszerűbb az elérhető jó kontraszt, az alacsony áramfelvétel és a dinamikus vezérelhetőség miatt. Az ún. STN (super twisted neumatic) mód ettől abban tér el, hogy a folyadékkristály molekulái a közrezáró lapokkal párhuzamosan helyezkednek el, és a polarizáló lapok tengelye 45 fokos szöget zár be a molekulák tengelyével, ez elősegíti a dinamikus vezérlést. A keresztülhaladó fény színe ebben az esetben az interferenciától függ. A fény színe megváltozik elektromos mező hatására. Ezalatt a molekulák tengelyállása 180 és 270 fok között változik. Az STN megjelenítésnél mód van a kék alapon történő fehér vagy sárga, illetve a fehér vagy sárga alapon történő kék színű megjelenítésre. A dinamikus vezérelhetőség jelentős növekedése miatt ez a fajta módszer alkalmas nagy kapacitású megjelenítők építéséhez, amilyeneket például a notebook gépekben találunk. Ebben a módban a függőleges felbontás 200-ra növelhető, míg a TN módnál ez maximálisan csak 100 lehet. Monokróm megjelenítés is lehetséges ebben a módban, ha egy optikai kompenzáló lapot iktatnak a polarizáló felületek közé. Ezt a módot nevezik NTN (neutralized STN) — azaz semlegesített super twist — módnak. Az STN mód másik változata az ún. FTN (formulated STN) mód, amikor egy speciális filmréteg helyettesíti a kompenzáló lapot. A folyadékkristály és a kompenzáló lap optikai paramétereit optimalizálva akár 1:20 vagy jobb kontrasztarány is elérhető. A dinamikus vezérlés szintén jelentősen, 400 fölé növelhető függőleges felbontást tesz lehetővé. A különböző dikroikus molekulák (áteső fényben különböző irányban különböző színű anyag) színe a molekulák elhelyezkedésétől függ. Ha ilyen dikroikus anyagot keverünk a folyadékkristályba, akkor annak molekulái az elektromos mező révén változtathatják helyüket.

A vezérlési módok

A fentiek szerint tehát a folyadékkristály optikai tulajdonságai elektromos mező hatására megváltoznak. A legalapvetőbb módszer a folyadékkristályos megjelenítés vezérlésére, amikor a

megjelenítő minden egyes pixeléhez egy-egy elektródapárt rendelünk, és az elektródák között feszültséget hozunk létre. Mivel a folyadékkristály károsodik, ha közvetlenül éri a feszültség, ezért egy váltakozó áramú négyszöghullámot kell a közönséges elektródához rendelni. Így a négyszöghullám azonos fázisában a pixel nem látható, az ellenkező fázisban pedig láthatóvá válik. A módszer hátránya a rengeteg elektróda, az ebből adódó huzalozási nehézségek és a külső áramkörökkel való kapcsolat építésének nehézkessége. E problémák megoldhatók a dinamikus vezérlési mód használatával, ami egyszerű mátrixmegjelenítést használ. Az elektródák sorokra és oszlopokra bontva helyezkednek el. Az alábbi két képlettel számítható ki a látható (Von) és a nem látható részeknél fellépő feszültség (Voff).

$$Von^2 = \frac{1}{T}\left[(Vs+Vd)^2\frac{T}{N} + \left(T - \frac{T}{N}\right)Vd^2\right]$$

$$Voff^2 = \frac{1}{T}\left[(Vs-Vd)^2\frac{T}{N} + \left(T - \frac{T}{N}\right)Vd^2\right]$$

A képletben Vs a sor-, Vd az oszlopelektródák feszültsége, N a pixelsorok száma, míg T az oszlopoké. Egy 640x200 képpontos felbontás esetén a látható és (Von) nem látható (Voff) pixelekre adott vezérlő feszültség 1,073 V.

LCD-nek áll a világ!

Elmondható tehát, hogy a folyadékkristályos megjelenítők alacsony energiaigényük, kis terjedelmük és tömegük miatt az elektronikus ábrázolás sok területén sikerrel alkalmazhatók. Bár kezdetben hátránnyal indultak a saját fényforrás hiánya miatt, azonban ezt az akadályt a háttérvilágítás (backlight) technikájával sikerült elhárítani. A nagy technikai variációs lehetőségek miatt az egyre tökéletesedő színes LCD megjelenítők lassan az íróasztalokra is felkerülnek az aktatáskákból és a mellényzsebekből.

Gáz van!

Bár a gázplazmás megjelenítők fő alkalmazási területe az ipar és a reklám, nem árt egy kis figyelmet fordítani rájuk. Több japán cég ugyanis a „falra akasztható" televízió sorozatgyártásának lehetőségét teremtette meg a gázplazmás megjelenítők fejlesztésével. A PDP megjelenítőknek kétféle típusa létezik: az egyenáramú és a váltakozó áramú. A plazmamegjelenítő egy önlumineszkáló rendszer, ami a gázban végbement elektromos kisülések fényét használja fel. A képernyő felépítése hasonló az LCD-éhez. Két, egymástól kb. 0,1 mm távolságra lévő elektródacsoporttal (X és Y) — melyek egymásra merőlegesen helyezkednek el — ellátott üveglap, melyek között (általában neon) gáz tölti ki a rést. Ha 100-200 voltos feszültség keletkezik az X és Y elektródák között, akkor a rést kitöltő gáz kisül, ami narancsszínű fényjelenséggel jár. A PDP megjelenítők a megjelenítés módja szerint két osztályba sorolhatók: a fix vagy közvetlen 8 szegmensű elektródákkal működő digitális és a sávelektródás, egymásra merőlegesen elhelyezkedő elektródákra építő grafikus megjelenítők osztályába, mely utóbbival alfanumerikus karakterek is megjeleníthetők.

Az AC/DC PDP-k

A váltakozó áramú PDP-k elektródái vékony szigetelőfilmmel vannak bevonva, és váltakozó áramot közvetítenek a gáz felé, míg az egyenáramú PDP-k elektródái közvetlenül adják le az egyenáramot a gáznak. Mivel a váltakozó áramú típusnál az elektródák nincsenek közvetlenül kitéve az ionbombázásnak, ezért megbízhatóbban és hosszabb ideig működnek. Az elektromos kisülés hatása miatt a szigetelőrétegen felgyülemlő töltés segítségével pedig egy megjelenítő memória működtetése válik lehetővé. A memória segítségével növelhető a megjelenítő mérete és kapacitása, a kontrasztra és a fényességre való kihatás nélkül. Emiatt a váltakozó áramú PDP-k jóval nagyobb teljesítményt nyújtanak.

Másrészt ebben az esetben a vezérlő áramkör jóval bonyolultabb, mivel töltésvezérlést is „rá kell ültetni" a vezérlő hullámformára. Az egyenáramú PDP-k vezérlő áramköre lényegesen egyszerűbb, és ennél fogva jóval olcsóbb. A módszer hátránya, hogy az elektródák és a kisülő gáz direkt érintkezése rontja a fényességet. Ennek oka, hogy egy előkisütés szükséges a stabil kisüléshez, melyben az adott terület nem bocsát ki fényt. Valamint a megjelenítő memóriahiány is oka a megnövekedett kapacitás melletti fényerőcsökkenésnek. A kontraszt javítása érdekében a gyártók fejlesztéseket folytatnak a szikrázásmentes elektródák kidolgozásának terén. A PDP-k általános tulajdonságai az alábbiakban foglalhatók össze röviden:

— Keskeny és könnyű kiépítés, torzulásmentes képpel (AC, DC)
— Önlumineszcencia, széles és tiszta megjelenítés (AC, DC)
— Nagy fényerő, éles kép
— Hosszú élettartam (AC)
— Gyors, néhány mikroszekundumos reakcióidő (AC, DC)
— Lehetőség az egy darabból álló nagyméretű megjelenítők építésére (elsősorban szabadtéri alkalmazás) (AC)
— Veszélyes sugárzások kizárhatósága (AC, DC)

Kis és közepes kapacitású PDP-k

A váltakozó áramú gázplazmás megjelenítőket használják például az olajszintmérő műszereknél és a nagyméretű információs paneleknél. Ez utóbbiaknál a szabadtéri alkalmazás és a nagy méret megköveteli a kontraszt és a fényerő növelését. Ezt a vezérlőfrekvencia csökkentésével és a lehető legfeketébb háttér alkalmazásával érték el. Bizonyos fajtákat kimondottan nappali használatra gyártanak. Az osztott képernyős megjelenítőknél, ahol viszonylag kis felület van — ilyenek például a jegyárusító automaták — a vezérléshez egyszerű frissítő áramkör is elég. Itt a memória használatának csak a fényerő növelése érdekében van értelme.

Nagy kapacitású PDP-k

A nagy kapacitású gázplazmás (640x400, 640x480) megjelenítők már

Folyadékkristályos képernyő (LCD, liquid crystal display)

1. Egyszerű mátrix LCD. Alacsony kontraszt, hosszú válaszidő, ezért mozgás követésére (játékra, prezentációra) nem alkalmas, de szövegszerkesztésre vagy adatbank kezeléséhez megfelelő. Vékony, könnyű, kisfogyasztású. Változatai: STN (super twisted neumatic), MSTN (monochrome STN), DSTN (double layer STN), TSTN (triple STN).

2. Aktív mátrix LCD. A plazmaképernyőhöz hasonló minőség, széles szürkeskála, ragyogó színek, éles kontraszt (100:1), gyors képváltás, csekély hőérzékenység. Lényege a vékonyréteg-technológia (TFT, thin film technology) alkalmazása. Minden képpont mögött a 3 alapszínnek megfelelő 3 FET tranzisztor (FET, field effect transistor) van, a hordozó üvegfelületre vékonyréteg technológiával felhordva. (Például 640x480 képpont esetén ez 921600 tranzisztort jelent.)

kimondottan a hordozható számítógépek piacára készültek, ahol kemény versenyben állnak az LCD megjelenítőkkel. Előnyük a lassabb LCD megjelenítőkkel szemben a nagy sebesség, mely az egérkurzor alkalmazásánál mutatkozik meg igazán. Ennek az ára azonban a nagyobb áramfelvétel és tömeg. Másik előnyük az LCD-vel szemben a nagyobb képernyőméret és a sokkal jobb fényerő és kontraszt. A laptoppiacon leginkább az egyenáramu PDP-k vannak jelen. A gyártók a nagyobb piaci részesedés érdekében igyekeznek a készülékek tömegét és az előállítási költségeket csökkenteni. A költségek csökkentéséhez elsősorban a magasfeszültségű vezérlő áramkörökön kellett változtatni. Ezt a gyártók a 160 voltos feszültséggel működő 64 bites LSI kifejlesztésével érték el. A tömeg csökkentéséhez szükség volt a gázt tartalmazó üveglapok karcsúsítására. Az egyenáramú típusoknál ez megoldható volt a panel részekre osztásával. Mára sikerült egy normál méretű PDP megjelenítő súlyát 1 kg alá csökkenteni. A költségek csökkentésére tenni kellett még valamit a vezérlő áramkörök előállításában is. A nagy kapacitású PDP-khez kifejlesztették a lebegőpontos vezérlési módot. A megjelenítés vezérlését szintén mátrixban (mint az LCD megjelenítők esetében), oszlopokba és sorokba rendezett elektródák segítségével oldották meg, ahol mindegyik kisütő elektródához egy-egy pár vezérlőelektróda tartozik. A fenti módszerrel a vezérlőáramkörök mérete a korábbiak tizedére csökkent. A megjelenítő memória segítségével pedig lehetővé vált, hogy az elektródák kevesebb mint 30 V feszültséggel működjenek.

Mostan színes PDP-ről álmodom...

Bár ez már valóság. A színes PDP megjelenítőknél a színek létrehozására a gázkisülések által ultraviola sugarakat kibocsátó, foszforalapú anyagokat használnak. A teljes színskála ebben az esetben fluoreszcens anyagok használatával érhető el a három alapszín (piros, zöld és kék) keverésével és különböző fényerejével. Az előbb említett nagy reakciósebességet és ezeket a színezési lehetőségeket ötvözve, és a nagyméretű, egy részből álló megjelenítés lehetőségével japán szakemberek eljutottak a „falra akasztható" HDTV gázplazmás alapú típusához. Ennek az üzletekben való megjelenésére azonban néhány évig még várnunk kell.

Tass Csaba

Adatátvitel

Gépek, ha találkoznak...

E számunk fő témája
a hordozható számítógépek alkalmazása.
Bár teljesítményük alapján
ezek a gépek egyre inkább helytállnak
egy átlagos felhasználási területen,
azonban használatuk még nem igazán általános.
A hordozható gépek tulajdonosait
a munkahelyen vagy otthon rendszerint egy nagyobb
— legalábbis méreteiben nagyobb — számítógép várja.
Örökös gond, hogy az adatok mikor, melyik gépen
találhatók meg, és hogyan kerülnek át egyikről a másikra.

Miért is tartunk otthon vagy a munkahelyen külön számítógépet a noteszgép mellett? Ha ezeket a gépeket csakis egy ember használja, és egyidejűleg nem is kell működniük, úgy az ok az asztali gép nagyobb teljesítménye vagy bővíthetősége kell legyen.

Elgondolkodtató, hogy a noteszgépek milyen teljesítményt érték el 1991 végére. Általános a 386SX mikroprocesszor használata 2-4 Mbájt memóriával, a beépített merevlemez kapacitása 40-80 Mbájt lehet. Bár a 3,5 inches floppymeghajtó rendszerint helyet kap az igencsak szűkre szabott készülékházban, ha 5,25 inches lemezre is szükségünk van, azt kívülről kell a géphez kapcsolni. A képernyő legtöbbször VGA-kompatibilis folyadékkristályos, egyszínű megjelenítő. Egyre kevésbé számít ritkaságnak a színes megjelenítő, ezek ára azonban jelenleg még meglehetősen borsos.

A számítási teljesítmény összemérhető a hazánkban leginkább elterjedt általános célú asztali gépek teljesítményével, legtöbbször meg is haladja azt. Amennyiben tehát egynél több kártyabővítő helyre nincs szükségünk, általában csak a megjelenítő és a billentyűzet minősége indokolja az asztali számítógép használatát.

Bármennyire is fejlődnek a noteszgépek képernyői, a felhasználók többsége ma még előnyben részesíti velük szemben a hagyományos katódsugárcsöves képernyőt, s mivel a billentyűzetek is általában a kényelmesen kezelhetőnél kisebbre sikerültek, sokan ezért is vonakodnak használni őket.

Az előzőekből adódik a megoldás, ugyanis a megfelelően kiválasztott noteszgép kényelmesen használható az íróasztalon is, ha beszerzünk mellé egy EGA- vagy VGA-felbontással rendelkező hagyományos monitort, és egy teljes méretű, 84 vagy 101 gombos billentyűzetet. A megoldás előnye és hátránya, hogy az adatok egy merevlemezen találhatók. Előnye azért, mert az ember igen feledékeny, és általában a legfontosabb esetekben felejti el az adatait átmásolni, hátránya azért, mert ez a merevlemez a rendszeres „utazás" következtében erősebben ki van téve a sérülésnek. Folyamatos és gondos adatmentési rendszert használva azonban a veszély csökkenthető.

Ha valamilyen okból kifolyólag mégis két gépen dolgozunk, megoldandó a gépek között az adatok cseréje. Ennek legegyszerűbb módja a floppylemezen történő „adatátvitel". Azonban még ennek a módszernek is vannak hardverkövetelményei, hiszen mindkét gépnek rendelkeznie kell egy azonos típusú floppymeghajtóval. Nagy tömegű adat rendszeres mozgatására a módszer nem javasolható a körülményessége és lassúsága miatt.

Ha a mágneses hordozót mint lehetséges adatátviteli eszközt elvetjük, marad az elektromos vagy fényjelekkel történő kapcsolatteremtés lehetősége. Munkahelyeken, ahol kialakították számítógépes helyi hálózatokat, nincs más teendő, mint a noteszgépbe beépített hálózati vezérlőkártyához csatlakoztatni az erre a célra fenntartott szabad kábelvéget.

A számítógép-hálózat kialakításának lehetőségével otthoni alkalmazás esetén nem kell igazán komolyan számolni. A legtöbb PC-kompatibilis számítógépben viszont megtalálható az a kártya, amellyel gépeinket összekapcsolhatjuk. Ez a kártya pedig az aszinkron soros vonali illesztő (COM1, COM2, ... port). A számítógép BIOS-ának leírásában azt olvashatjuk, hogy a soros illesztőkártyával legfeljebb 9600 baud (bit/másodperc) sebesség érhető el. Ilyen sebességgel egy 100 kbájt méretű állomány továbbítása ellenőrzéssel együtt 2-4 percig is eltarthat, ami nem túlságosan vonzó adat.

A BIOS megkerülésével azonban a soros port jóval magasabb (mintegy 115 kbaud!) sebességre is programozható, és ezt néhány később ismertetendő segédprogram meg is teszi. Az átvitel történhet szoftverszinkronizálással, ekkor az illesztőkábel a lehető legegyszerűbb, 3 eres, úgynevezett null-modem kábel. Hardverszinkronizáció (handshake) alkalmazásával az elérhető sebesség nagyobb lehet, ez azonban 6 vagy 7 vezetéket tartalmazó drágább kábelt igényel.

A gépek párhuzamos (nyomtató) illesztőkártyája is alkalmas lehet adatátvitelre. Mivel a port hardverkialakítása nem teszi lehetővé a 8 bit szélességű adatbeolvasást, a párhuzamos illesztés valójában csak 4 bites. Az elérhető sebesség a soros átvitel sebességét 25...400%-kal haladhatja meg, erősen függ a gép sebességétől.

Megkönnyítendő két PC összekapcsolását, számos segédprogram született. Ezek mindegyikének tartalmaznia kell az átvitt adatok hibamentességének ellenőrzését. Az adatátvitel rendszerint csomagokban történik. Minden csomag fejléccel, törzzsel és farokkal rendelkezik. A fejléc tartalmazza a csomag azonosítóját, hosszát és egyéb más információt.

A törzs maga a tényleges adatblokk, majd ezután következik a kiszámított hibaellenőrzési kód. A törzs hossza befolyásolja az átviteli sebességet, hiszen jó minőségű zavarmentes vonalon a hosszabb törzs ritkábban kívánja meg redundáns információ átvitelét. Az egyes programok kezelőszervei között elvétve találhatunk közvetlenül a csomagméretet beállító parancsot, azonban a <Turbo> fedőnevű opció gyakran ezt változtatja.

Az adatátviteli hibák felderítésére az adó minden egyes továbbított csomagban elküldi a tényleges adattörzs vagy az egész csomag számított hibaellenőrző kódját. Amennyiben a vevő hibát detektál, ezt jelzi az adónak, majd a csomag ismételt továbbítása következik. Az ellenőrző kód alapvetően kétféle lehet, ellenőrző összeg (checksum) vagy ciklikus redundanciakód (CRC). Az előbbi az egyszerűbb, gyorsabban előállítható, míg az utóbbi alaposabb védelmet nyújt.

Szokás kombináltan használni ezeket, esetleg automatikus választással. Ez utóbbi működhet például úgy, hogy az adatátvitel checksum ellenőrzéssel indul, és hiba nélküli esetben így is fejeződik be. Az első hiba előfordulása után azonban az adó és a vevő egyaránt átkapcsol CRC ellenőrzési üzemmódba, védekezve az esetleges többszörös hibák ellen.

A PC-felhasználók egyik kedvenc segédprogramja, a Norton Commander 3.0 verziószámú változata is tartalmazza a gépek összekapcsolásának lehetőségét. Az egyik panelen a Link opciót kiválasztva utasíthatjuk a gépeket a kapcsolat felvételére. A kapcsolat aszimmetrikus, azaz csakis az egyik gépen adhatunk ki parancsokat. A parancsok a „szolga" gép lemezegységeire, azok állományaira vonatkozhatnak, másolást, törlést, alkönyvtár-létrehozást engednek meg. A kapcsolat bontását a mestergép kezdeményezheti, felépítését azonban mindkét gépen be kell billentyűzni, ami esetleg kényelmetlen lehet.

A csatlakozó kábel 3 eres null-modem kábel. Az átvitel sebessége 57 vagy 115 kbaud lehet. A Turbo mód kikapcsolása a karakterek közti szünetek idejének megnyújtásával növeli az átvitel biztonságát a sebesség csökkenése árán.

A Norton Commandernek nem a soros kapcsolat az erőssége, nem is erre koncentráltak a megvalósítása során, azonban egy nagyon egyszerű, azonnal rendelkezésre álló adatátviteli segédeszközt adtak a felhasználók kezébe.

Természetesen kimondottan két számítógép összekapcsolására készült, igen komoly lehetőségeket nyújtó programok is beszerezhetők a szoftverpiacon. Ezek közül két, igen hasonló képességű terméket említek meg, a Fastwire és a Laplink programcsomagot.

A Fastwire elegáns, egyszerűen kezelhető, menüvezérelt program. A segítségével felépített kapcsolat olyannyira aszimmetrikus, hogy a két gépen különböző programot kell elindítani. Viszont a három említett program közül egyedül rendelkezik azzal a lehetőséggel, hogy a mester gépről a távvezérelt gépen programot lehessen elindítani, bár a kommunikációs program ekkor mintegy 70 kbájtot elfoglal az amúgy is mindig kevés memóriából.

A Fastwire működhet null-modem kábellel, 7 eres soros, valamint párhuzamos kábellel egyaránt. Az átviteli sebesség a kábelek felsorolásának sorrendjében növekszik, azonban a null-modem kábellel is elérheti a 115 kbaud értéket.

A DOS megrögzött parancssoros felhasználóinak jó hír, hogy a Fastwire átkapcsolható parancsmódba, ahol használható az operációs rendszer megszokott parancskészlete. Rendelkezésre áll viszont 3 új parancs, amellyel állományokat másolhatunk a gépek között, valamint parancsot hajtathatunk végre a távoli gépen.

Az állományok átvitele sokrétűen paraméterezhető, például lehetséges csak a másik gépen levőnél fiatalabb vagy bizonyos dátum után keletkezett állományok továbbítása. Makrókat is szerkeszthetünk, amelyek segítségével olyan összetett feladatok végezhetők el néhány billentyűleütéssel, mint egy teljes lemez áttöltése a nagyobb tárolókapacitású másik gépre.

A Laplink programot készítői — mint neve is mutatja — kimondottan laptop és asztali számítógép összekapcsolására szánták. Szolgáltatásai színvonalában nagyon közel áll a már ismertetett Fastwire programcsomaghoz. Alkalmas soros (null-modem és hardver handshake-kábellel egyaránt) és párhuzamos adatátvitelre, a sebességadatok hasonlóak.

Egyedülálló jó tulajdonsága, hogy a gépek közti kapcsolat szimmetrikus, azaz mindkét gép billentyűzetéről vezérelhető az adatok mozgatása. Természetesen állománycsoportok is kijelölhetők, amelyek átvitele nem igényli a kezelő beavatkozását.

Meglepő szolgáltatás, hogy a Laplink installálható a soros kábelen keresztül, akár floppyegység nélküli számítógépen is. Ehhez mindössze 2 DOS parancsot kell beírnunk, a többi teendőt távvezérelten elvégzi a program.

Összességében elmondhatjuk tehát, hogy ha valaki a laptopját vagy notebookját kívánja rendszeresen együtt használni egy asztali géppel, úgy érdemes a Fastwire vagy a Laplink programok valamelyikét megvásárolnia. Ugyanez igaz olyan esetekre is, ahol két gépen dolgoznak, és hálózatot kialakítani nem érdemes. Úrvezetőknek, hétvégi kirándulóknak, tehát akiknek csak ritkán van szükségük adattovábbításra, a Norton Commander is jó segítőtárs lehet.

Lóth Tamás

Áramforrások

Magától nem megy — még!

A hordozható áramforrásokról — amelyek vegyi-elektromos átalakításon alapulnak — mindenkinek az autókban jól ismert savas ólomakkumulátorok jutnak eszébe. Sajnos ezek térfogatuk és a használat során fejlődő gázok miatti kötelező nyitott kivitelük alapján nem igazán hordozhatók. A jelenlegi hordozható számítógépekben a leggyakrabban alkalmazott áramforrások a nikkel-kadmium akkumulátorok. Tipikusan 3-4 órát képesek folyamatosan működni töltés nélkül, néhány százszor tölthetők újra, mielőtt cserélni kell őket.

Van néhány ellenérv az akkumulátorok használatával kapcsolatban. Az első egy környezeti szempont: a kadmium nagyon mérgező anyag, így a használt akkumulátorok kidobása veszélyes lehet a környezetre. A másik a töltésével kapcsolatos. Az akku „emlékszik" a kisütöttségi fokára. Amikor töltés után ismét eléri ezt a kisütöttségi szintet, feszültsége akár 10%-kal is lecsökkenhet, ami azt jelzi a töltő áramkörnek, hogy az akku lemerült. Azaz a helyes kisütési-töltési eljárás során szinte a teljes kisütés után kell újra tölteni, hogy a kapacitását megőrizze. (Nem helyes tehát az akkumulátoros berendezésen rajta hagyni a töltőt.)

A közelmúltban publikált fejlesztési eredmények több új alternatívát is kínálnak. Ilyenek a Gates Energy Systems cég által fejlesztett nikkel-hidrid akkumulátorok, amelyek 50%-kal nagyobb teljesítményt tudnak szolgáltatni azonos térfogat mellett. Egy más megoldást kínál az Aerobic Power Systems által fejlesztett levegő-ón akkumulátor.

A tápellátás kezelése és szabályozása

A tápellátás kezelésével kapcsolatos megoldások a kémia, a fizika, az áramkörtervezés és programozás eredményeit hasznosítják. Ma már nyugodtan kijelenthetjük, hogy egy termék sikerét a táplálásának magas színvonalon történő megoldása is befolyásolja. Ez azt jelenti, hogy a készüléket működtető áramforrásoknak hosszú időn keresztül, stabilan kell biztosítaniuk a készülék működőképességét. A felhasznált teljesítmény minden wattját gondosan be kell osztani, és úgy kell takarékoskodni vele, ahogy ez lehetséges. Ehhez a berendezés működésében meg kell különböztetni három üzemállapotot: működési állapotban a berendezés és részegységei működnek, nyugalmi állapotban az egységek bekapcsolva vannak, de nem aktívak (például a floppyegység nem forog), „szundi" állapotban pedig a részegységek a lehető legkisebb energiát fogyasztják.

A következőkben egy kicsit részletesebben is tárgyaljuk a kérdést, és illusztrációként három „siker-noteszgépet", az Atari Portfoliót, a Hewlett-Packard HP95LX-ét, és a Poqet Computer cég Poqet gépét hozzuk fel példaként. Azért választottunk ilyen noteszgépeket, mert ezeknél a fogyasztás minimalizálása még élesebben merül fel a méreteik miatt.

Hova megy a teljesítmény?

Az információ feldolgozásához igen kevés energia szükséges. Mint ahogy ez könnyen becsülhető, a betáplált energia legnagyobb része hővé alakul a berendezés áramköreiben. Egy kisebb része a mozgó alkatrészeket tartalmazó egységekben — diszkmeghajtókban — mechanikai energiává alakul, egy elenyésző része fénnyé és elektromágneses sugárzássá.

Elektromos energiával táplált berendezésekben két egyszerű egyenlettel írhatjuk fel a felszabaduló teljesítményt.

Egyenáramú táplálás esetén a P = U*U/R kifejezés alkalmazható. Azaz a berendezést tápláló feszültség (U) négyzetével egyenes, a készülék ellenállásával (R) fordított arányban van a készülékben felszabaduló teljesítmény (P).

Mikor egy váltóáramú jel — például órajel — táplál valamilyen áramkört, akkor P = C*U*U*f összefüggés használható. Azaz az áramkör kapacitásával (C) az órajel frekvenciájával (f) és a feszültség négyzetével arányos a veszteség. Ezekre az egyenletekre a későbbiekben is hivatkozni fogunk.

A feszültség változtatása

Mindkét egyenlet mutatja, hogy a tápfeszültségtől jelentősen — négyzetesen — függ a fogyasztás. Ezért a fogyasztáscsökkentés egyik módja lehet, ha csökkentjük a feszültséget. A valóságban számos integrált áramkör már használja ezt a megoldást, bár ez az adatlapokból nem világlik ki. Ez azért van így, mert a külvilág felé „5 voltra termettnek" láttatja magát, de a belső áramkörei kisebb tápfeszültséggel működnek.

Ha ilyen, csökkentett feszültségről működő áramkörökből építjük fel a rendszert, akkor a működtető tápfeszültséget is kisebbre választhatjuk. Például az Atari Portfolio csupán 4,5 V-ról működik (3 darab 1,5 V-os AA elem), ami a négyzetes függés miatt 20%-os fogyasztáscsökkenést ad.

Az újabb fejlesztésű áramköröknél még jelentősebb fogyasztáscsökkenés várható. Az Intelnek van olyan 80186-os verziója, amely 3 V-ról működik, és gyárthatók olyan tisztán CMOS áramkörök, amelyeknek 2 V is elegendő. Az igazi korlát az, hogy minden összekapcsolt résznek működnie kell ilyen kis feszültségen.

Statikus terhelések

Az áramkörtervezők igen sokszor kötnek a táp és a földpont közé ellenállást.

Ezek azok, amelyeket más szóval feszültségosztó, felhúzó és lehúzó ellenállásoknak hívunk. Feladatuk egy adott pont potenciáljának a biztosítása. Ezek állandóan fogyasztanak, ugyanakkor helyettesíthetők olyan aktív kapcsolásokkal, amelyeknek a fogyasztása töredéke az ellenállásos megoldásnak. Például a feszültségosztót, amelynek nagyobb áramot kell szolgáltatnia, kisebb ellenállású — és ezért sokat fogyasztó — részekből kell összállítani. Ez a megoldás helyettesíthető egy igen nagy ellenállásokból — és ezért keveset fogyasztó — összerakott osztóra kapcsolódó műveleti erősítővel is, amelynek kimenete szolgáltatja a megfelelő áramot. Természetesen az ilyen megoldások költségesebbek, mint a csak ellenállásokat alkalmazók, de a fogyasztáscsökkenés haszna ezt kiegyenlítheti.

A fenti második egyenlet szerint a fogyasztás a számítógépek órajellel táplált áramköreinél a frekvenciával arányos. A jelenleg használt mikroprocesszorok többségének belső regiszterei dinamikus működésűek, azaz tartalmuk megőrzése érdekében folyamatos frissítést igényelnek. Ilyen esetben a működtető órajelet csupán egy bizonyos határig lehet csökkenteni. A megoldás az, hogy az órajel leállításakor el kell menteni ezen frissítést igénylő belső regiszterek tartalmát, majd az órajel megjelenésekor visszaállítani. A gyakorlatban ezt a technikát használják az AMD 286LX és az Intel 386SL CPU áramköreinél.

A kijelző...

A megjelenítő egyszerűen a legtöbb energiát fogyasztó egység. Az ideális kijelző olyan lenne, amit az eredményjelző táblákná már használnak: egy impulzus hatására az adott képpont állapotot vált, és újabb impulzus csak az újabb állapotváltáskor szükséges.

Sajnos ez a fajta — statikus — kijelző jelenleg még kísérleti stádiumban van, és még túl lassú is. Ezért jelenleg minden hordozható gépben a leggyakrabban a folyamatos frissítést igénylő LCD kijelzőket használják.

A kijelző mérete meghatározó a fogyasztásban. Nagyobb képernyő használatakor, a villódzás elkerülése miatt a képernyőt meghajtó órajel frekvenciáját a sorok számával arányosan növelni kell. Mivel minden képpont egy kis kapacitás töltését-kisütését jelenti, ezért ha rövidebb az idő, a kapacitást töltő feszültséget kell megnövelni.

A Portfolióban belső felhúzó ellenállásokat is tartalmazó Hitachi áramkör és egy karakter ROM hajtja meg a 10 sor x 40 oszlopos kijelzőt. Emiatt nagyon gyorsan lemerülnek az elemek.

A HP 95LX-ben külön tervezett áramkörök és nagy értékű ellenállásokból álló osztó vezérli a 16 sor x 40 oszlopos kijelzőt; a képernyőt frissítő jel frekvenciáját is 70-ről 50 Hz-re csökkentik telepes táplálásnál.

A jelenlegi legjobb megoldást a Poqetben találhatjuk: analóg meghajtó áramköröket használ az aktív feszültségek előállítására, statikus terhelés nincs. Ezért van az, hogy a 25 sor x 80 oszlopos kijelzőkkel is hetekig képes elemcsere nélkül működni.

Az LCD kijelzőknél esetleg szükséges háttérvilágítás jelentős fogyasztást jelent. Ezért a notesz PC-knél még csak ritkán alkalmazzák, a laptopoknál a vevők igényei miatt használják.

A memória frissítése

A gyakorlatban kétféle RAM memóriát használnak: a nagy áramköri sűrűségű, de sokat fogyasztó, periodikus frissítést igénylő dinamikus RAM-ot, és a keveset fogyasztó, de drága és terjedelmesebb statikus RAM-ot. Létezik a kettő közötti ún. kvázistatikus RAM, ahol a tokban lévő belső frissítő áramkör végzi el a dinamikus memóriacellák frissítését.

A lassú frissítéssel ritkábban történik meg a cellák frissítése. Ez bizonyos DRAM típusoknál lehetséges. Az elosztott frissítés alkalmazásakor a memóriatokokat egyenként frissítik, a nagyobb teljesítményigényű, egyszerre történő frissítés helyett.

A RAS előtti CAS: számos DRAM típusnál használható ritkán alkalmazott módszer. A tokban lévő belső számláló szolgáltatja a frissíteni kívánt memória mátrix oszlopainak címét. Ez kisebb fogyasztást jelent annál, mint ha külső címzőrendszer generálná ezeket a címeket.

A ROM csökkentése

A ROM az, ahol a rendszertervezők sok teljesítményt takaríthatnak meg. Először az Apple II-ben használták azt a megoldást, hogy a perifériakártyákon lévő ROM memóriák egy kapcsolótranzisztoron keresztül kaptak tápfeszültséget. A ROM-ok csak a tényleges működésük idején kaptak táplálást.

A modern számítógépek már általánosan használják ezt a módszert. A shadow-RAM módszernél a viszonylag lassú ROM-ban tárolt tartalmat átmásolják a dinamikus RAM memóriába. A módszernek további előnyeit is kihasználják:

— a másolás után a ROM táplálása kikapcsolható,

— a BIOS tartalmát az olcsóbb 8 bites ROM-okból másolhatják át,

— a BIOS a ROM-ban tömörített formában lehet, amit áttöltéskor „csomagolnak ki".

Diétára fogott áramkörök

A tápellátás kezelésénél általános szabály, hogy lehetőleg minél kevesebb önálló áramkörből kell a gépet felépíteni. A külvilág felé a tokok általában sokat fogyasztó és a belsőnél nagyobb tápfeszültségen működő puffereken keresztül kapcsolódnak.

A megoldás: magasan integrált áramkörök alkalmazása. A 286LX áramkör egy AT számítógép egyetlen tokban. A 386SL és 286SL tokok mindegyike is majdnem egy komplett számítógép.

A Chips and Technologies, Headland Technology/Cirrus Logic, VLSI Logic és más cégek is teljes áramkörkészleteket gyártanak. Ezek mindegyike áramköri támogatást is biztosít a teljesítmény csökkentéséhez.

A 386SL áramkör több szempontból is figyelemre méltó. A szabványos AT perifériaáramkörök integrálása mellett egy új megszakítást is bevezettek: a rendszerkezelő megszakítást (System Management Interrupt = SMI). Az SMI, amely kiváltható bármely teljesítménycsökkentést célzó eseménnyel, még a konvencionálisan legnagyobb prioritású NMI megszakításnál is nagyobb prioritású. Mikor egy SMI bekövetkezik, a processzort a 386SX valós módjához hasonlóan egy máskülönben elérhetetlen címtérre állítja. Ilyenkor a processzor speciális időzítőkkel, regiszterekkel és jelzőbitekkel manipulálhat, amely a teljesítménnyel való gazdálkodást segíti. A gyártó, az Intel szerint ennek a legnagyobb előnye, hogy ez a fajta működés a használt operációs rendszertől független.

A sokat fogyasztó perifériák

A képernyőn kívül még más perifériák fogyasztását is számításba kell venni.

A soros portok sok energiát fogyasztanak bekapcsoláskor. A szabványos RS232 kimenetek 9000 ohmos ellenálláson keresztül tartják fenn a +/–12 V-os feszültséget. Néhány megoldásnál ehelyett a 0 V/+5 V-os jelszinteket használják, de ez kompatibilitási problémákat vet fel. Továbbá a soros adatátvitelt megvalósító UART áramkörnek saját

kvarcgenerátora van, a szabványos adatátviteli sebesség megvalósításához. A laptopoknál nem ez a fogyasztás a meghatározó, de a tenyérnyi gépek, mint a HP95LX és a Portfolio elemei gyorsan lemerülnek, ha a soros port be van kapcsolva.

Miért nem kapcsolja le a gép maga automatikusan a soros portot? Sajnos, például a ROM-BIOS-ban nagyon szerényen megvalósított sorosport-kezelő esetén sincs külön parancs vagy lehetőség a soros port megnyitására; a futó program egyszerűen elkezdi használni, amikor kell.

Ilyen külön megnyitási parancs nélkül a gép csak „sejtheti" a soros port automatikus kezelését. Ilyen „sejtő" megoldás lehet például, hogy a portot a gép kikapcsolja, ha bizonyos ideig a soros vonalon nem jött adat. Ez azonban hatásában azzal a kellemetlenséggel jár, hogy például egy szövegszerkesztőben történő pár perces gépelés után az egérhez nyúlva az nem működik.

Százszázalékos megoldás nincs a soros vonal fogyasztáskezelésére. A fogyasztást kezelő program figyelheti az adatátvitelt jelző „Carrier Detect" és „Data Terminal Ready" vonalakat, amelyek jelezhetik a vonal használatát. Ha egy egérmeghajtót installálnak, a szoftver felismerheti, és hívást generálhat az egér újrainicializálására, ha a port ki volt kapcsolva. A HP 95LX-ben a soros port alaphelyzetben ki van kapcsolva, míg egy beépített alkalmazói program fut. Akkor kapcsolódik be, ha a DOS programok futnak, de ilyenkor is kézileg kikapcsolható.

A belső modemek is alacsony fogyasztású üzemmódban vannak, amíg nem érkezik a hívás. Néhány modem magáról a telefonvonalról veszi a működéséhez szükséges tápfeszültséget.

A billentyűzetek maguk is önálló mikroszámítógépeket tartalmaznak, így megvan a lehetőség az automatikus „szundizásra". Ilyen mikroprocesszor az Intel 80C51SL gyárilag beírt programmal rendelkező áramköre. A HP 95LX „házilag tervezett" vezérlőt tartalmaz, és a billentyűzetet nem folyamatosan figyeli. Csak egy billentyűnyomásra vár, és utána feléledve azonosítja, hogy melyik volt.

A hajlékonylemezek minden rendszerben a legtöbbet pihennek, de az Intel bejelentett egy új, kis fogyasztású, floppyvezérlő áramkört, a 82077-es típust.

A legtöbb programban a merevlemez használata elég ritka esemény, így a meghajtót és vezérlőjét ki lehet kapcsolni. Nagyon jelentős a betáplált energia átmérőtől való függése: ezért a cél a minél kisebb méretű merevlemezmeghajtók felhasználása.

A nyomtatóport is kikapcsolható, de ha például hardverkulcs (dongle) van rádugva, az problémát jelenthet.

Min lehet még spórolni? Ha kicsi a fogyasztás, nem kell a tápegységet hűtő ventilátor sem. A különféle jelzésekre használt világító diódákat sem tanácsos alkalmazni, mert áramfogyasztásuk jelentős. Végül, sok rendszer a hangszóró kikapcsolását is lehetővé teszi, ami ismét teljesítménymegtakarítással jár.

Az üresjárás felismerése

A legnagyobb technikai kihívás annak pontos megállapítása, hogy a rendszer mikor „nem csinál semmit", tehát alacsony fogyasztású, szunnyadó módra kapcsolható. A Poqet, a Portfolio és a HP 95LX beépített programjai ezt pontosan jelzik, a többi hagyományos programnál erre nincs pontos azonosítás. A hordozható gépek gyártóinak így külön algoritmusokat kellett kidolgozniuk az üresjárás felismerésére. Az ilyen algoritmusoknak pontosaknak kell lenniük: ha nem azok, akkor a telep élettartama jelentősen csökken.

Néhány DOS program aktivitását a keretben mellékelt táblázat mutatja.

A Digital Research BatteryMax programja (a DRDOS-szal együttműködve), a Phoenix Technologies Miser BIOS-a és a Poqetben lévő programok példázzák az üresjárás figyelését. Mindegyik figyeli az éppen futó programot, hogy az használja-e a billentyűzetet. Ha a program állandóan a billentyűzetet figyeli, vagy egy olyan BIOS-hívást aktivizál, amelyikből a billentyű lenyomás nélkül nem tér vissza, akkor valószínűleg billentyűre vár. A CPU ilyenkor lelassítható vagy megállítható, amíg billentyűnyomás nem érkezik.

A DOS Idle megszakítása (INT28H) azt jelzi, hogy a DOS egy karaktert vár a billentyűzetről, a get line (egy sor beolvasása) függvényben. Vagy ha egy program a futó időt kérdezi folyamatosan, nagy valószínűséggel valami késleltetést futtat. Ilyenkor a figyelő rendszer általában bizonyos számú hívás után lecsökkenti a rendszeraktivitást.

Sajnos a helyzet azért nem ilyen egyszerű. Például az INT28H nem mindig jelenti azt, hogy a rendszer szundizhat. Egy számolótábla például hosszabb számításoknál ellenőrzi a billentyűzet megnyomását, hogy nem akarja-e a felhasználó megszakítani a számítást. A legtöbb kommunikációs program felváltva figyeli a billentyűzetet és egy állapotjelzőt a memóriában, ami a beérkezett karakter okozta megszakítást jelzi. A TSR programok is meg tudják bolondítani a figyelést azzal, hogy folyamatosan figyelik a rendszeridőt, azt mutatva, mintha semmi hasznosat nem csinálnának.

Ezért a teljesítménykezelő programnak „okosnak" kell lennie. Például tudnia kell, hogy a számolótáblánál a számítás közbeni billentyűzetfigyelés ritkább. A kommunikációs programoknál is eldönthető, hogy volt-e már előzőleg a soros port aktív, valamint az adatátviteli sebességet is a maximumra lehet állítani az aktív idő rövidítése céljából. A TSR program is kivehető-visszatölthető annak eldöntésére, hogy a főprogram aktivitását meg lehessen különböztetni a TSR programétól.

A képernyőhöz, diszkekhez és soros portokhoz fordulást a programaktivitás biztos jeleként lehet értékelni.

És a következtetés...

Végül is a fogyasztás csökkentésére irányuló kutatások és eredményeik mindenkinek hasznosak. A kevesebbet fogyasztó áramkörök — mivel hőmérsékletük alacsonyabb — megbízhatóbban működnek, mint a villanyrezsóként viselkedők. A régebbi, telepről táplált „buta" eszközeinket felváltják a magas beépített intelligenciával rendelkező eszközök. Nincs már túl messze az a világ, amelyben a gépek mindenhol elérhetők és használhatók anélkül, hogy azokat a fali csatlakozóba dugnánk.

Kónya László

Néhány DOS-program aktivitása

Program	Művelet	Üresjárás, %-ban
COMMAND.COM	Bevitelre várakozás	99,4
	DIR, TYPE, CD parancsok	55,9
Számolótábla	Bevitelre várakozás	88,1
	Szerkesztés, mozgás	67,5
Szövegszerkesztő	Bevitelre várakozás	91,3
	Gépelés, nyomtatás	89,3

soknál. A figyelmes olvasó bizonyára könnyen el tudja különíteni ezeket az adatokat. (Ilyen adat például az egyik gépnél: a „zavarszűrés" hiányaként utalunk arra, hogy teljesen szokatlan módon ez a gép rádiófrekvenciás kisugárzással zavarja környezetét. A „továbbfejlesztés" 1-es érdemjegye pedig arra utal, hogy a cég már lemondott a modell továbbfejlesztéséről, kifutottnak tekinti ezt a géptípust.)

— A „globális érték" összesen csak 5 gépnél szerepel — azoknál, amelyeket a tesztelést végző szakemberek a legmagasabbra értékeltek minden szempontból.

Vizsgáljuk meg először, hogy menynyiben tesznek eleget a jogos elvárásoknak az olcsóbb árfekvésű noteszgépek. A „picik" között is a legkisebb és a legkönnyebb az első kettő, a CompuAdd és a Northgate jól használható kis apróságai. Az elsőnek gyenge pontja a képernyő, a másiknak viszont iszonyúan gyorsan kimerül a telepe. Sokkal jobban jár, aki a Dell gépét veszi meg, amelynek ugyan kisebb a képernyője, de minden paramétere jó, áramtakarékosságban pedig csak egyetlen gép előzi meg, az ugyancsak az olcsó kategóriába tartozó Austin.

A közepes méretűek között az első kettő, a Leading és a Blackship lényegében ugyanazt nyújtja, csak a Leading magasabb árért. A Blackship a mezőny legolcsóbb gépe, üzemeltetése is gazdaságos, igaz viszont, hogy a teljesítménye gyenge. A harmadik, a Micro ugyancsak igen olcsó, sokat kibír, de ennek is a teljesítményével vannak problémák.

A nagyobbacskák közül az Everex viszi el a pálmát. Szinte minden szempontból kitűnő gép, és nagyszerűen bírja a strapát. (A Panasonickal együtt ez a gép volt az, amely minden gyötrést kibírt.) Az Austin képernyője gyengébb, viszont áramtakarékossága kiváló. A Zeos elég szélsőséges konstrukció. Sok jó tulajdonsága van, de főleg teljesítményben lemarad az Everextől (igaz, olcsóbb is nála).

Háromezer dollár felett

Egyszerűség kedvéért (és mivel viszonylag kevesebb gép került a magasabb árkategóriákba) egy táblázatban foglaltuk össze a közepes és drága árfekvésű gépek értékelő adatait. Több egyszerűsítést is végrehajtunk a táblázatban (így például a teljesítményen belül nem látszik szükségesnek a CPU és a merevlemez teljesítményének különválasztása, mivel értékelésük között nincs lényeges különbség). Az összevont táblázaton belül azonban megtartjuk az alkategorizálást a homogenitás biztosítása érdekében.

A középdrága kategória „pici" gépei közé került a Texas Instruments kis könnyű TravelMate gépe, amely szinte minden paramétere szerint az „olcsó pici" CompuAdd ikertestvére lehetne. Csak lényegesen magasabb áron. A szintén olcsó Delltől meg éppen lemarad áramtakarékosságban és a képernyő teljesítményében. Az Acer már lényegesen jobb konstrukció, de kis mérete ellenére rendkívül nehéz, és ez is eszi az energiát.

A Hyundai viszonylag olcsó, igen masszív konstrukció, de számos gyenge pontja van. Különösen a billentyűzete kiábrándítóan gyatra. Kimondottan jó noteszgép a Tandon, bár nem olcsó. Zavaró a billentyűk kényelmetlen elhelyezkedése. A rázkódás kivételével jól tűri a gyötrést.

A „nagycsoportos" középdrága kategóriában a Panasonic átlagos teljesítményt nyújt (elmarad például az olcsó Delltől), viszont a „kínzóteszt" összes viszontagságait a teljes mezőnyből ez a típus viselte el legkönnyebben (a másik hasonlóan rendkívül masszív konstrukció az Everex volt, amely amellett, hogy sokkal olcsóbb, mint a Panasonic, jobb képernyővel és billentyűzettel van felszerelve).

A drága gépek kategóriájában a „picik" közé tartozik a Zenith — alig nagyobb az „ikerpárnál", a CompuAddnél és a Texas Instruments TravelMatejénél. Igen jó teljesítményt nyújt, de rendkívül drágán. Különlegesen jó teljesítményének fő oka, hogy ez már az energiatakarékos Intel 386SL mikroprocesszorral van felszerelve — az első fecske az SL noteszgépek közül. Kis méretéhez képest súlya rendkívül nagy.

Teljesítményben messze lemarad tőle a Librex, fogyasztásban viszont jócskán megelőzi. Bár elvileg van energiatakarékos üzemmódja, ez nem működik. Az energiafaló Northgate után ez a legdrágább üzemeltetésű noteszgép. 4 MB-os RAM-ja kevés vigaszt jelent egyéb opcióinak gyengesége miatt.

Egyelőre nem ismerjük még az új Toshiba T2200SX noteszgépnek az árát, de kisöccséről, a nagyobbacska T2000SX-ről ítélve igen jónak ígérkezik. Mérete annyira lecsökkent, hogy jelenleg ez a legkisebb és legkönnyebb noteszgép, alig 2,5 kg. Intelligens áramellátó automatikával van felszerelve, így fogyasztásban is alighanem veri az egész mezőnyt. Remélhetően árban is versenyképes lesz vetélytársaival.

A „kicsik" közé tartozik a drága kategóriában két kiváló energiatakarékos noteszgép, az AT&T Safarija és a Compaq LTE-je. Mérnöki szempontból a Compaqot tekintik ma a szinte tökéletes noteszgépnek. Teljesítménye, képernyőjének minősége semmi kívánnivalót nem hagy maga után. A nyúzótesztet is kibírta, még a bekapcsolt állapotban való leejtést is, bár a belső modemmel adódtak némi problémák.

	Középmezőny						Drágák				
	„Picik"		„Kicsik"		„Nagycs."		„Picik"		„Kicsik"		„Nagycs."
	a	b	a	b	a	c	b	c	a	b	a
	TM	Acer	Hyun	Tand	Panas	AST	Zen	Libr	AT&T	Comq	Tosh
Használhatóság I. Alapkövetelmények											
Teljesítmény	5	4	4	5	4	5	6	4		7	5
Képernyő	1	6		3	3	2		2		7	4
Olvashatóság									3		3
Billentyűk		3	1			2	2		6		3
Elrendezés				2						2	
Kényelem										6	
Masszivitás	5	3	5	5	7	3		6	3	7	5
Használhatóság II. További szempontok											
Portok		7	2						6	6	
Konfigurálás	5			4							
Setup							5				
Opciók		7	2	6			5	2	6	6	5
Belső modem	+	—	—	+	+	+	+	+	+	+	+
2MB min. RAM	+	—	—	+	—	+	+	41	+	+	+
BIOS setup								3			
Vezérlés							6	3			
Tűrőképesség											
Rázkódás	5	5	5	4	5	5	4	5	1	4	5
Hideg	4	4	5	5	5	4	5	3	5	3	4
Meleg	5	3	5	5	5	5	5	5	—	5	3
Kávéteszt	4	5	4	5	5	4	4	5	—	5	5
Esés csukva	5	1	5	5	5	1	1	5	—	3	5
Esés nyitva	1	—	1	1	3	—	—	1	—	3	1
Kényelmi szempontok											
Súly	7	2	3	3		2	3	5		2	
Méret	7	5		5	2	2	6	5		3	2
Stílus			2						7		
Külcsín		6									
Költségek											
Ár			4			4	1		1	1	3
Fogyasztás	3	2			4		5	2	6	6	5
Takarék üzem						1		1			
Takarékos SL							7				
Globális érték		4		4							

A Safariról sok egyéb kiválósága mellett elmondható, hogy megvan a maga elegáns működési stílusa. Egyébként energiatakarékosságban is élen jár, csak az Austin előzte meg. (Az Austin több mint 7 óráig bírta, a Safari jó 5 és fél óráig.)

Elérkeztünk a drága gépek „nagyobbacskáihoz". A Compaq színvonalát ugyan nem éri el, de nagyon megbízható, jó konstrukció a Toshiba T2000SX. Legfőbb hátránya a viszonylag nagy méret. Fentebb láttuk már, hogy ezzel a Toshiba is elégedetlen volt, és elkészítette — helyette? mellette? — a legkisebb noteszgépet. Hibájának róhatjuk még fel, hogy latin betűs karakterei nem elég jól olvashatóak. (Ebben a hibában egyébként osztozik az AT&T Safarijával.)

Utoljára hagytunk egy egészen frissen megjelent új noteszgépet, amely szintén a „leg"-ek közé tartozik. A NEC néhány hónapja jelentette be UltraLite 386 SX/20-ának a piacra dobását. Energiatakarékos megoldása mellett igényes paramétereivel méltó versenytársnak ígérkezik az Everex és a Compaq mellett. Sajnos, még valamiben a „leg"-ek közé tartozik. Ez ideig ez a legdrágább a 386SX-alapú noteszgépek között...

Vargha Dénes

Egy „bombabiztos" ipari laptop

Kívül kicsi, belül nagy

Bizonyos helyzetekben a laptop egyben maga „a" munkahelyi számítógép. Gondoljunk csak arra, amikor a gépnek eleve állandóan úton kell lennie, gépkocsiban, hajón vagy repülőn. Másrészt gondoljunk a bontási-építkezési terepekre, vagy mindazon ipari környezetekre, ahol nem lehet kiküszöbölni a port, füstöt, vegyi anyagokat, hőhatást, rázkódást, ezért a helyszínen számítógépet eddig nem is nagyon alkalmaztak, pedig már szükség lett volna rá.

A fenti körülmények között végzendő számítógépes munkára fejlesztette ki a német Kontron Elektronik az IP Lite hordozható számítógépet. Ennek a 10 kilónyi vállra akasztható berendezésnek általános mutatói megfelelnek a normál méretű „jobbfajta" gépekének. Legigényesebb változatában 486-os processzor, 33 MHz-es órajel, 16 megabájtig terjedő RAM „ketyeg", és 40, 100 vagy 200 megás merevlemezzel szerelhető fel, másik lemezegységébe pedig a szokásos 3,5 collos floppy való. Képernyője 256 tónusú, egyszínű VGA, háttérvilágítással, 180 fokig folyamatosan állítható keretben.

Az IP Lite különlegessége leginkább a működési biztonságát garantáló extrákban fedezhető fel. Működés közben 4 G, szállításkor pedig 15 G gyorsulást is elvisel. Ebben nagy szerepe van a magnéziumöntvényből készített fémháznak, de az önmagában kevés lenne. Az ütődés és a vibráció ellen egyes érzékeny alkatrészeket külön is védeni kell. A por és a levegőben lévő többi szennyező anyag távol tartására a gépben szűrőrendszer és állandó túlnyomás van. A ventilátor fordulatszáma nagyobb külső hőhatást érzékelve automatikusan emelkedik, egy kritikus hőmérsékleti határnál pedig a gép kikapcsolódik. Beépített készülék figyeli a főbb belső szerkezeti elemek áramellátását, és zavart észlelve azonnal „intézkedik". Igen jó a gép védelme a külső elektromos zavarok ellen is.

Figyelmet érdemlőek e nem mindennapi ipari laptop „felfegyverzésének" lehetőségei: gépkategóriájában világviszonylatban egyedülállóan EISA adatsínnel rendelkezik, s hét db 32 bites bővítési dugaszoló helye közül 5 szabad. Külső csatlakozói: 2 db soros (ebből az egyik átkapcsolható RS 422/485-re), 1 párhuzamos, 1 egér, továbbá az „áramőr" jelkimenete. Ezeken kívül még egy színes „vendégmonitornak" és egy külső meghajtónak (5,25"-es floppynak vagy streamernek) is van külön csatlakozója.

Ennek a számítógépnek van viszont egy apró szépséghibája. Bizonyosan sokan kitalálták, hogy mi lehet az: természetesen az ára. Amit a minőségért mindig meg kell fizetni. Bár belegondolva, hogy a színes képernyőjű laptop gépek 20 ezer márka alatt ma még nem szerezhetők meg, nem is tűnik olyan soknak az IP Lite-ért az a 15 ezer márka.

Faklen Pál



Szakirodalmi válogatás a hónap témájához

Angol nyelvű cikkek

IBM and AT&T enter the fray of 386SX notebook computers. (Az IBM és AT&T új műszaki megoldásokkal jelentkező 386SX-alapú, 20 MHz-es táskagépeinek bemutatása.)
Byte, 1991/8.
Full color comes to LCDs. (Hordozható számítógépek szabványos VGA módú színes folyadékkristályos megjelenítővel.)
Byte, 1991/8.
NCR knows notepads. (Az NCR System 3125 fényceruzás számítógép ismertetése.)
Byte, 1991/8.
386SX laptops: desktop power notebook size. (27, 16 és 20 MHz-es, 386SX-alapú notesz számítógépek átfogó értékelése benchmark teszteredményeik alapján.)
PC Magazine, 1991/14.
Notebooks: keeping in the swim. (Öt 386SX- és egy 386SL-alapú noteszszámítógép bemutatása és értékelése.)
What Micro? 1991/8.
First 386SL PC arrives. (Piacra került az első 386SL-alapú táskagép, a Zenith Mastersport 386SL.)
What Micro? 1991/7.
Palmtop PCs: power by ounce. (Hat notesz-számítógép bemutatása és értékelése.)
PC Magazine, 1991/13.
Put an XT in your pocket. (A Hewlett-Packard 95LX notesz számítógépe.)
PC Computing, 1991/5.
Battle of the laptop stars. (Hat 386SX-alapú noteszszámítógép összehasonlító tesztje.)
PC Computing, 1991/5.
Eden develops first U.K. pen-based computer. (Az Eden Group angol cég PC-DOS-kompatibilis 386SX-alapú fényceruzás számítógépe — Papertalk VPi386.)
Byte, 1991/5.
No compromise notebooks with 386SX power. (Nyolc DOS, illetve DR DOS alapú, 386SX-es hordozható táskagép bemutatása és benchmark teszteredményei.)
Byte, 1991/6.
Software to link laptops to desktops. (Táskagépek számára készült adatállomány-átviteli programok.)
Datamation. 1991/9.
Preview: IBM notebook worth the wait. (Az IBM PS/2 Model L40 SX — 386-os alapú, speciális szoftverrel működő, 20 MHz-es új táskagépe.)
PC World, 1991/5.
MicroSlate introduces keyboardless notebooks. (A MicroSlate billentyűzet nélküli notesz számítógépe.)
Infoworld, 1991/19.
Tandon ships 20-MHz 386 SX notebook computer. (A Tandon 20 MHz-es 386 SX táskagépe.)
Infoworld, 1991/16.
Psion 400 portable swaps DOS compatibility for speed, light weight, proprietary software. (A Psion 400 könnyű, részben DOS-kompatibilis táskagépe.)
Infoworld, 1991/11.
IS adjusts as portables fly the coop. (A noteszgépek használata az informatikáért felelős vezetők szemszögéből.)
Computerworld, 1991/16.
How to buy. (Vásárlási tanácsadás hordozható PC-k és különféle perifériák beszerzéséhez.)
What Micro? 1991/5.
Fax modem chips bring datacom to laptops. (Hordozható személyi számítógépekbe beépíthető fax modem chipek az adatok és képek átvitelére.)
Electronic Design, 1991/6.
Lightweight portable PCs. (23 könnyű súlyú, hordozható számítógép összehasonlító értékelése teszteredményeik alapján.)
Which Computer? 1991/4.
Sony's Unix to go. (Sony NEWS 3250: Unix rendszerű táskagép.)
Datamation, 1991/6.
Grid and Go attack different market sectors with pen-based machines. (Növekvő piaci igények a fényceruzás bevitelt alkalmazó hordozható gépek iránt.)
Computer Age, 1991/4.

Német nyelvű cikkek

Laptop im Einsatz. (Laptopok alkalmazása.)
Chip, 1991/7.
Mit Windows unterwegs. (A Windows használata laptop gépeken.)
Computer Persönlich, 1991/16.
Vergleich: 286-Laptops. (Kilenc, 7000 márkánál nem drágább, 286-os táskagép értékelése benchmark tesztek alapján.)
Chip, 1991/5.
Weggefährten: 18 Notebooks bis 9000 Mark. (18 táskagéptípus összehasonlító tesztelése.)
Computer Persönlich, 1991/16.
Der mobile Computer an der Dockstation. (Dokkállomások hordozható személyi számítógépek számára.)
Sysdata, 1991/6.
Im Osten was Neues. (Négy — 4900 és 6300 márka közötti áron kapható — táskagép összehasonlító értékelése.)
Computer Persönlich, 1991/11.
Ratgeber: Tragbare Computer. (Mire ügyeljünk a hordozható számítógép vásárlásakor?)
PC Praxis, 1991/7.
Übersicht: Tragbare PC. (A 2000 és 22000 NSZK márka közötti hordozható személyi számítógépek piaci kínálata.)
Chip, 1991/5.
Alles am Griff. (Húsz hordozható, táska- és noteszszámítógép összehasonlító értékelése.)
Computer Persönlich, 1991/7.
Der Laptop 'liest'. (Penpoint operációs rendszer laptopokba fényceruzával történő adatbeviteléhez.)
PC Welt, 1991/5.
Mehr Pep als Schlepp: 36 tragbare Computer unter 3,5 kg. (3,5 kg-nál nem nehezebb táska- és noteszszámítógépek piaci kínálata.)
MC Mikrocomputer-Zeitschrift, 1991/2.

Magyar nyelvű cikkek

CHIP-teszt: tizenkét szék.
Chip, 1991/10.
IBM PS/2 Laptop.
Computerworld-Számítástechnika, 1991/36.
PS/2 notesz és társai.
Computerworld-Számítástechnika, 1991/30.
Előre a tajvani úton.
Computerworld-Számítástechnika, 1991/29.
IBM PS/2 P75: Koffer a köbön.
Computerworld-Számítástechnika, 1991/24.
Heten, mint a noteszok.
Computerworld-Számítástechnika, 1991/13.
Sharp PC-6200.
Chip, 1991/1.

Az összeállítás a Sandokan adatbázis alapján készült.
InfoNet Kft.
1119 Budapest
XI., Vahot u. 6.
Telefon: 166-9065
infoNET

Adat- és programtárolás

Emlék-elmék

Az előző részekben átnéztük, hogyan is működik a számítógép két legfontosabb kommunikációs perifériája, a billentyűzet és a monitor. A most következő részekben az adat- és programtárolás témakörére térünk át. Értelemszerűen két, minőségileg különböző tárolási rendszerrel foglalkozunk. Az egyik az operatív memória, a másik pedig a mágneslemezes berendezések köre. E havi számunkban főleg az operatív memóriáról lesz szó, de érintjük a floppyk és winchesterek lelkivilágát is.

A memóriák

A számítógépekben alapvetően kétféle memória-áramkört találhatunk. Az egyik a ROM memória, amelynek tartalmát nem tudjuk megváltoztatni. A ROM memória a gép kikapcsolt állapotában is megőrzi tartalmát. A ROM memóriák tartalmát a gyártó cég beírja a chipbe, és az többé nem változtatható meg. A másik memóriatípus a RAM. Ez a memória csak a gép bekapcsolt állapotában működik, kikapcsolás után tartalma elvész. A RAM memória tartalmát szabadon megváltoztathatjuk, átírhatjuk.

Az IBM PC gépen futó DOS operációs rendszer maximum 1 megabájt memória kezelésére képes. Ennek az 1 megabájtnak igazából csak az alsó fele áll a DOS rendelkezésére, mert a felső felét a képernyővezérlő, a winchester- és floppyvezérlő, valamint a ROM memóriába „égetett" alapprogram, a ROM BIOS foglalja el. Így igazából a DOS részére 512 kbájt RAM memória marad szabadon (XT gép esetén 640 kbájt).

A RAM típusú memória-áramkörök egy része az alapkártyán található, de lehetőség van plusz memóriakártya behelyezésére is. Programozási szempontból azonban mindegy, hogy a memória fizikailag hol helyezkedik el, csak az számít, hogy mennyi van beépítve a gépbe, és hogyan használhatjuk azokat.

A DOS a rábízott 512 kbájt memóriával gazdálkodhat. Ennek egy részét a BIOS kezeli, egy részét a DOS belső célokra használja fel, egy részét pedig a programok vehetik igénybe. A memória legalján a 0:0 (szegmens:offset) és 0:3FF címek között helyezkedik el a megszakítási vektortáblázat. (Lásd a rovatindító, októberi cikkünkben.) A közvetlenül e feletti címeken 0:400 és 0:4FF között a BIOS garázdálkodik, ezt a DOS adatterülete követi 0:500 és 0:5FF között. Ezeket az alsó memóriatartományokat mindig az itt feltüntetett címeken találhatjuk, de az ezek után következők pontos helye már a használt DOS-verziótól és -paraméterektől függ. Annyi biztos, hogy a 0:5FF címen végződő DOS-terület felett az IO.SYS (vagy IBM-DOS esetén az IBMBIO.SYS) található, ennek a hossza már verziófüggő. Közvetlenül ezután következik az MSDOS.SYS (illetve IBM-DOS esetén az IBMDOS.SYS). Felettük helyezkednek el az alapértelmezett vagy konfigurációs állományban megadott device driverek (eszközmeghajtók), például az ANSI.SYS, VDISK.SYS, MOUSE.SYS stb. A legfelső (legutoljára betöltött) eszközmeghajtó felett található a COMMAND.COM. Az ezután szabadon maradó területből utal ki a DOS a programok számára memóriát, ha erre szükség van.

A programok elindításának első lépése a betöltés az operatív memóriába. Ez általános esetben a tár legalacsonyabb szabad részére való beolvasást jelent. A beolvasott program a teljes szabad tárterületet megkapja az operációs rendszertől, nem csak annyit, amekkora helyen elférne. A rendelkezésére álló helynek azt a részét, amelyet saját kódja nem foglal el, szabadon használhatja. Blokkokat allokálhat, overlay ágakat olvashat be stb. Kilépés után a rendelkezésre bocsátott memóriaterületet általában visszaadja az operációs rendszernek, kivéve a renitens rezidens programokat. Ezek úgy adják vissza a vezérlést a DOS-nak, hogy közben bizonyos méretű memóriát (legalább a rezidens kód tárolására valót) megtartanak maguknak. Az ábra egy lehetséges memóriaelrendezést mutat be.

A DOS minden egyes memóriablokk előtt elhelyez egy 16 bájtnyi leíró területet. Ennek a leírónak a létrehozása és módosítása a DOS kizárólagos joga! A leíróterület első bájtja azt mutatja meg, hogy az adott blokk a legfelső (utolsó)-e a blokkok sorában vagy nem. 5A(hexa) az utolsó blokk kódja, 4E(hexa) az összes többié. A következő két bájt a

Egy lehetséges memóriatérkép

Felhasználói memória	Rendszerfüggő
COMMAND.COM	Rendszerfüggő
Memóriarezidens programok	Rendszerfüggő
MSDOS.SYS vagy IBMDOS.SYS	Rendszerfüggő
IO.SYS vagy IBMBIO.SYS	0000:0600 — ?
DOS-munkaterület	0000:0500 — 0000:05FF
BIOS-munkaterület	0000:0400 — 0000:04FF
Megszakítási vektorok	0000:0000 — 0000:03FF

blokk tulajdonosát azonosítja, azt a programot, amelyik a blokkot lefoglalta vagy birtokolja. Ez a két bájt a tulajdonos program tárban elfoglalt helyére mutat. A leíró terület következő két bájtja a blokk paragrafusokban (16 bájt) kifejezett hosszát mutatja. Ebbe a hosszba nincs beleszámolva a leíróterület 1 paragrafusa (16 bájtja).

Háttértárolók

A RAM és ROM memóriák egyik nagy előnye a nagyon gyors műveletvégzési lehetőség és a könnyű kezelhetőség. Hátrányuk ezzel szemben a viszonylag kicsi kapacitás, a magas ár és az a tény, hogy az operatív memória RAM áramkörei a gép kikapcsolásakor mindent elfelejtenek.

A winchesterek használata sokkal lassabb ugyan, mint a memóriáé, de kapacitásuk a memóriáénak jó esetben 80-100-szorosa is lehet. A winchester a tárolt adatokat a gép kikapcsolt állapotában is megőrzi.

A floppylemez kapacitása is nagyon kicsi, sebessége is lassú, de cserélhetősége miatt mégis szükség van rá.

A floppy- és a winchesterlemezek (továbbiakban: mágneslemezek) belső felépítése és működési módja teljesen azonos. A nagy különbségek abból adódnak, hogy a winchester — néhány kiviteltől eltekintve — nem cserélhető, egy gyárilag lezárt dobozban, abszolút pormentes körülmények között dolgozik, míg a floppykat a szabad levegőn tároljuk, és néha (véletlenül) meg is kínozzuk.

A mágneslemez úgynevezett blokkos periféria. Ez azt jelenti, hogy az adatokat nem bájtonként írjuk ki vagy olvassuk be, hanem meghatározott méretű bájtcsoportokat (blokkokat) kezelünk. Ha a mágneslemezen tárolt adatok közül csak egy bájtot szeretnénk megváltoztatni, akkor is egy egész blokkot be kell olvasni, az adott bájtot átírni, majd az egész blokkot visszaírni a lemezre. A mágneslemezes blokkok mérete egy nulla és öt közti számmal jellemezhető (jele: N). A blokk méretét bájtokban a 128*2^N képlet adja meg. Az N = 0 érték 128 bájtos, az N = 5 érték 4096 bájtos szektorméretet jelöl. N értéke valójában ötnél nagyobb szám is lehet, de ez már nem szabályos.

A mágneslemezek kör alakú műanyag korongok, amelyek külső felülete mindkét oldalon vas-oxiddal vagy más hasonló mágneses tulajdonságú anyaggal van bevonva. Erre a mágneses rétegre írja fel a meghajtóegység az adatokat mágneses jelek formájában. A vas-oxid a mágneses jelet megőrzi, és szükség esetén a meghajtó vissza tudja olvasni a felírt adatokat.

A floppylemezeknek mindig két oldaluk van, a winchesterlemezeknek viszont általában ennél több (4, 6, 8, vagy akár 10). Ezt úgy érik el, hogy egy winchester igazából nem egy, hanem több lemezből áll. Egy öt lemezből álló winchesterlemeznek például értelemszerűen tíz oldala van.

A mágneslemez felületén az adatok kör alakú sávokban (track) vannak felírva. Az egy mágneslemezre írható trackek száma floppylemezeknél 40 és 100 között mozog, winchestereken elérheti az ezret is. Az egyes trackek az adathordozó felületén kijelölt kör alakú sávok. Ezek a sávok még tovább vannak osztva körcikkelyekre, úgynevezett szektorokra. Egy szektornyi az a legkisebb (és legnagyobb) adatmennyiség, amit egyszerre olvasni vagy írni tudunk. Ha ennél kevesebb adatra van szükség, akkor is kénytelenek vagyunk beolvasni egy egész szektort, ha többre, akkor pedig több szektort kell egymás után beolvasni. Egy track floppylemezek esetén 8 és 20 közötti szektort tartalmaz, winchestereken ez a szám maximum 32 lehet.

Azért óvatosan!

Az eddig leírtakból következik, hogy egy adott szektor azonosításához meg kell adnunk a következő három dolgot: 1. melyik lemezoldalon van a szektor (head vagy side), 2. az adott oldal melyik sávjában van (track), és végezetül 3., hogy az adott sávon belül hányadik szektorról van szó (sector).

A floppylemezek minden egyes szektora a hasznos adaton kívül egy csomó járulékos információt tárol. A szektorok felépítése a következő: 12 db 0, 3 db 161, 254, tracksorszám, headsorszám, szektorsorszám, szektorméret-jelzőbájt, 2 bájt ID mező CRC, 16 db 78, 12 db 0, 3 db 161, 250, szektorhasznos adatok, 2 bájt CRC, 78 értékű „kitöltő" bájtok a következő szektor elejéig.

A felsorolásban használt kifejezések: „tracksorszám", „headsorszám", „szektorsorszám" a fentebb említett három szektorazonosító információ. A szektorméret-jelzőbájt a fentebb részletezett N értéket tartalmazza. A DOS által használt lemezeken itt mindig 2 áll, mert ez adja az 512 bájtos szektorméretet (a DOS minden mágneslemezen 512 bájtos szektorméretet használ). Az ID mező CRC két bájtja az eddig ismertetett adatokról készített adathelyesség-ellenőrző számot tárolja. A hasznos adatok után következő két CRC bájt a hasznos adatok helyességét ellenőrző számokat tárolja.

Az egyes sávok elején az első szektor előtt még egy bájtsorozatot találhatunk, amely a sáv kezdetét azonosítja. A bájtok a következők: 50 db 78, 12 db 0, 3 db 194, 252, 32 db 78. Ezt sávfejlécnek nevezhetjük. A sávfejléc után következnek az egyes szektorok, amelyek belső felépítését az előző bekezdés tartalmazza.

A kontroller egyszerre mindig egy szektornyi információt kezel, és a szektornak csak a hasznos adatokat tartalmazó részét, valamint az ehhez tartozó CRC-információt közli velünk. A sávfejléc és az egyes szektorok fejléce mindig rejtve marad. Ezt a szabályt egy ötletes módon tudjuk csak megkerülni. Ha lemezovasási művelet előtt a szektorméret jelzőbájtjába a szokásos érték (2) helyett kilencet írunk, akkor a szektorméretet a rendszer 16 384 bájtnak fogja venni. A szektor hossza ettől ténylegesen nem változik. Ha most ezt a szektort beolvassuk, akkor az olvasás nem áll meg a szektor végén, hanem folytatódik a szektor után következő CRC-vel, majd a szektorok közötti „kitöltő" bájtokkal, a következő szektorral, és így tovább.

Ezeknek a lehetőségeknek a kihasználásával nagyszerű programvédelmet lehet készíteni. A mágneslemezek fizikai kezelését kizárólag profi programozóknak is csak akkor ajánlom, ha részletes és hiteles leírás áll a rendelkezésükre.

Fridl György

Lebegő matematika koprocesszorra

PC-vásárláskor a gyanakvóbb felhasználó általában — mit adnak a pénzemért alapon — belenéz a gép dobozába, és meglátván egy üres foglalatot a processzor közelében, ritkán állja meg kérdés nélkül:
— Mit spóroltak el innen?
„A koprocesszor helye" — mondja az eladó. További firtatásra esetleg még hozzáteszi: „A lebegőpontos számításokhoz. Opcionális". Vagyis gépünk enélkül is működik. Mi kerül ezen ennyibe, és egyáltalán mit tud? Erre a kérdésre igyekszik válaszolni négyrészes cikksorozatunk, amihez azonban először némi alapozás szükséges.

A felhasználó, főleg ha szövegszerkesztésre használja PC-jét, többet nem is találkozik vele. Ha azonban például táblázatkezelésre is szüksége van, meglepve tapasztalhatja, hogy ugyanaz a program egy barát vagy egy kolléga gépén sokkal előbb végez a számításokkal. „Koprocesszorom van." — mondja büszkén a kolléga. Ha pedig valamelyik CAD programot kívánja használni, előfordulhat, hogy az el sem indul a koprocesszor — más néven matematikai processzor — nélküli gépen. Néhány közülük még egy hibaüzenetre sem méltatja a komolytalannak ítélt felhasználót.

A második meglepetés akkor éri, ha a fentiek hatására úgy dönt, beszerzi ezt az izét. Ára ugyanis eléri — típustól és sebességtől függően — a teljes konfiguráció (winchesterestül-monitorostul) árának 20-30%-át.

A processzorok fejlődésével növekedett a regiszterekben ábrázolható számok nagysága is, de ez még mindig nem elegendő nagyobb üzleti, tudományos és műszaki alkalmazásokhoz. Természetesen a programozó tetszés szerinti nagyságú számokat használhat, de ezek kezelése nehézkes és lassú. További problémát jelent, hogy a PC-kben használt processzorok utasításszinten csak a négy alapműveletet tudják végrehajtani. Minden más függvényt olyan függvénysorokból kell előállítani, amelyekben csak ez a négy művelet szerepel. (Ami viszont nagyon sok időt vesz igénybe, és a kód méretét is erősen megnöveli.) A számítások hatékonyságának fokozására alkalmazzák a matematikai processzorokat, amelyek lényegesen nagyobb méretű számokat kezelnek, és utasításszinten valósítanak meg több magasabb rendű függvényt. A velük szemben támasztott minimális követelményeket az IEEE 754-1985-ös szabványa tartalmazza.

A szabvány szerint a nagyméretű számokat a papír-ceruza módszernél megszokott normál alakban ábrázoljuk. Ekkor a szám értékes jegyeit az első rész, a mantissza, míg nagyságrendjét a második tag, a kitevő tartalmazza. A szabvány háromféle lebegőpontos formátumot definiál, az egyszeres és a kétszeres, valamint a kétszeres kiterjesztett pontosságú valós alakot. Mindhárom formátum előjel-, kitevő- és mantisszamezőt tartalmaz. A normál alakú számábrázolásnál az 1-nél kisebb számok kitevője negatív. Mivel a formátumban az előjel a mantisszára vonatkozik — negatív szám is ábrázolható —, a kitevőhöz hozzáadnak egy eltolásértéket. Az eltolás értékénél kisebb kitevő negatívként értelmezett.

Nézzünk egy példát, hogyan ábrázoljuk a matematikai processzoron belül mondjuk a –245.625-öt.

Először is bináris számmá kell átváltanunk:

–245.625 = –11110101.101B

Ezután a tizedespontot el kell léptetnünk balra, az első értékes jegyig, ami helyi értékenként kettővel való osztást jelent, és hogy a szám értéke helyes

1. ábra

Az IEEE 754—1985 szabvány számformátumai

Mezők				Formátum	Tartomány
S	Kitevő, 8 bit	Mantissza, 23 bit		Egyszeres pontosságú valós	0, 1.2E—38<x<3.4E38
S	Kitevő, 11 bit	Mantissza, 53 bit		Kétszeres pontosságú valós	0, 2.3E—308<x<1.7E308
S	Kitevő, 15 bit	1	Mantissza, 64 bit	Kétszeres kiterjesztett pontosságú valós	0, 3.4E—4932<x<1.1E4932

S = előjelbit
S = 1 negatív
S = 0 pozitív

A kitevő eltolása: 127 az egyszeres, 1 023 a kétszeres, 16 383 a kétszeres kiterjesztett formánál.

maradjon, ezt a kitevővel kell ellensúlyozni:
–1.1110101101 E 111B
(mantissza) (kitevő)

A fenti műveletet nevezik normálásnak. Egynél kisebb szám esetén a tizedespontot jobbra kell léptetni, amit negatív kitevő ellensúlyoz. A normálásra azért van szükség, mert a mantisszán belül nem tudjuk jelölni a tizedespont helyét, így azt egy fix, megállapodott pozícióra kell állítanunk, és legalább egy értékes jegye minden számnak van. Az első értékes jegy egy bináris számnál mindig csak egyes lehet, hiszen a nulla nem az, ha nem áll előtte semmi. Így tulajdonképpen a tizedespont előtt álló egyes elhagyható, ezután minden szám részének tekintjük, viszont ezzel a fix hosszúságú mantisszában nyertünk még egy ábrázolható bitet.

Ezután már csak az eltolást kell hozzáadnunk a kitevőhöz, ami az egyszeres pontosságú valós alak esetén 127.
111 + 01111111 = 10000110

A szám végső alakja az egyszeres pontosságú formában:
1 előjel
10000110 kitevő (8 bit)
11101011010000000000000 mantissza (23 bit)

Pontatlanság

Az 1. ábra mutatja a szabvány által definiált formákat, és az egyes típusokban ábrázolható számtartományokat. Ezek nagysága láttán azonban nem szabad elfelejteni, hogy a számok csak korlátozott pontossággal ábrázolhatók. Pontosságon két, egymást követő és ábrázolható szám távolságát értjük a számegyenesen (lásd a 2. ábrát). Ha egy tartományon csak korlátozott — a mantissza hossza által megszabott — számú érték ábrázolható, akkor a pontosság nyilván nem lehet tetszőleges.

Nagyobb gondot okoz, hogy a pontosság a szám nagyságától is függ.

Ennek illusztrálására nézzünk egy példát, amelyben a mantissza és a kitevő is csak két számjegyet tartalmaz. Az egyszerűség kedvéért maradjunk a tízes számrendszerben, és tekintsünk el a negatív számoktól, valamint a normálástól.

A kétjegyű mantisszán 1 és 99 között minden egész számot ábrázolhatunk, ekkor a kitevő 0.
mantissza — kitevő
01 00, azaz $1x10^0 = 1x1 = 1$,
99 00 $99x10^0 = 99x1 = 99$.

A 100-at is tudjuk ábrázolni:
10 01, azaz $10x10^1 = 10x10 = 100$.

A 101-gyel azonban már baj van, három értékes jegyet tartalmaz, ábrázolni azonban csak kettőt tudunk, ha csonkítjuk, a szám értéke elvész. A 100 után következő, pontosan ábrázolható szám a 110.
11 01, azaz $11x10^1 = 11x10 = 110$.

A pontosság (azaz pontatlanság) értéke 10.

A háromjegyű számoknál az 1000 után már csak az 1100-at tudjuk ábrázolni, a pontosság 100 lesz. A relatív — a szám nagyságához viszonyított — pontosság nem változott, 1 százalékos (mivel két értékes jegyet használtunk tízes számrendszerben), az ábrázolás abszolút pontossága azonban romlott.

Ha egy számítás eredménye két ábrázolható szám közé esik, akkor kerekíteni kell, a kerekítésekre vonatkozó előírásokat szintén tartalmazza a szabvány.

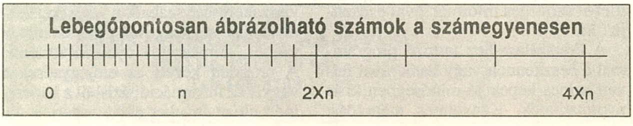
A fentiek miatt az erősen eltérő nagyságrendű számokkal végzett műveletek váratlan eredményekre vezethetnek. Ha egy nagy számhoz egy nála lényegesen kisebb számot adunk vagy vonunk ki, előfordul, hogy a szám változatlan marad.

Így például, ha koprocesszort használunk, a
FOR i=0 TO 1000000000 STEP 1
ciklusnak sohasem lesz vége, hacsak nem gondoskodunk külön róla, hogy a lépésérték növelje a számlálót.

Miért nem használjuk — a pontosság növelésére — csak a legnagyobb, a kétszeres kiterjesztett pontosságú formát? Méreténél fogva a betöltése több mint kétszer annyi ideig tart, és a memóriában is több helyet foglal, kis számoknál felesleges 0-kat tartalmazva. E hosszúra nyúlt bevezető után a következő részben a speciális értékek problémakörének boncolgatásával folytatjuk a sorozatot.

Csórián Sándor

2. ábra



FrameBase vagy Kerestext?
Van képük hozzá!

A hazai piacon több képi és szöveges adatbázis-kezelő rendszer látott napvilágot. Ezek közül most két hazai fejlesztésű rendszert — FrameBase és Kerestext 2.01 — szeretnénk nagyító alá venni.

FrameBase

A FrameBase egy olyan színes képi adatbázis-kezelő program, amely kép- és szövegfájlok archiválására, megjelenítésére és visszakeresésére alkalmas. Ugyanakkor a program tárolja azokat a kapcsolatokat, amelyeket a felhasználó definiál a képek és a szövegek között. A program a képi információkat TIFF-, a szövegeseket pedig ASCII-formátumban tárolja.

A rendszer indításakor egy konfigurációs fájlban rögzítjük a képfelbontásra és a háttértárra vonatkozó információkat, és itt választjuk ki a program menüjének és üzeneteinek a nyelvét (magyar, német, angol). A képek és a hozzá tartozó szövegfájl úgynevezett kulcsszavakkal (címekkel) kapcsolhatók össze. A tárolt információkat ezen kulcsszavak alapján rendszerezhetjük és kereshetjük vissza. A kulcsszavak között definiálhatunk logikai kapcsolatokat, amelyek jelentősen megkönnyítik a visszakeresést.

A FrameBase a nemzetközi TIFF-szabványoknak megfelelően tárolt kétszintes, tónusos fekete-fehér, valamint 256 színes kép megjelenítésére alkalmas. A képek természetesen kicsinyíthetők, nagyíthatók, forgathatók. Kiinduláskor a kép kicsinyítve jelenik meg a képernyőn. A felhasználó gyakorlatilag addig nagyíthatja a képet, ameddig 4 képpont teljesen kitölti a képernyőt. A nagyítás után a képernyőt „úsztatni" lehet a kép felett, így a kép minden részletét megnézhetjük.

A FrameBase TIFF-szabványoknak megfelelő tömörítő eljárásokat is tartalmaz. Ezek a tömörítések veszteségmentesek, az egyes képek bonyolultságának megfelelően 30-70% hatásfokúak lehetnek. Tekintettel az adatok nagy mennyiségére, célszerű nagy kapacitású tárakat használni, így a FrameBase támogatja az optikai diszkek háttértárként való alkalmazását.

Hypermap-hypertext keresés

Az adatbázisban való keresés jól bevált módszere a hypermap-hypertext keresés. Ez azt jelenti, hogy a felhasználó kijelölheti a kép vagy a szöveg egy részletét, és a kulcsszó definiálásával nevet ad ennek a részletnek. A kurzorral a kép vagy a szöveg egy részletére rámutatva olyan címlistához jutunk, amely a választott kulcsszó által meghatározott osztály elemeit tartalmazza. Ezek közül egyet választva az eljárás tovább folytatható. A FrameBase ismertetésekor erre látványos példát mutattak be a fejlesztők. Először a spanyol tengerpart jelent jelent meg a képernyőn. Majd rákattintva egy kiválasztott tengerparti városra, utána rögtön választhattunk az üdülőhely szállodái közül. Végül a kijelölt szálloda szobáit is „bejárhattuk" pillanatok alatt. Ez lehet az oka, hogy az idegenforgalom területén például szállodai helyfoglaló rendszerként vagy a falusi turizmus nyilvántartására már több helyen is használnak FrameBase-t.

Mivel a rendszer a kulcsszavakat karaktersorozatként értelmezi, így numerikus kulcsok szerinti keresés közvetlenül nem lehetséges. Ezt a hiányosságot azonban a fejlesztők azzal hidalták át, hogy a FrameBase-t alkalmassá tették más adatbázis-kezelő rendszerekkel való együttműködésre, azaz paraméterek átadására, illetve fogadására. Így a már létező adatbázisok állományait a FrameBase-zel további képi, illetve szöveges információval egészíthetjük ki.

A FrameBase-hez tartozó programmal a beszkennelt vagy kamerával felvett színes képek jó minőségben ki is nyomtathatók. Kényelmes megoldás, hogy valamennyi nyomtatóhoz kifejlesztették a megfelelő nyomtatóprogramot.

Alkalmazási területek

A FrameBase-t ma már több helyen használják Magyarországon, hiszen a múzeumi, az áruházi, az alkatrész-, autó- és bútorkatalógusok összeállításakor vagy az orvosi felvételek és metszetképek rendszerezésében jelentősen megkönnyíti a felhasználó munkáját. Egy érdekes, izgalmas felhasználói terület a rendőrségi alkalmazás, ahol egy arckép adatbázisból a szemtanú választja ki a szemet, orrot, fület, szájat. Az így előállított fantomképhez természetesen ujjlenyomat és aláírás is tartozhat.

A DOS alatt működő, C-ben megírt program méltányos, 90 000 forintos (+ÁFA) árával valószínűleg a keresett programok közé tartozik majd, annak ellenére, hogy van egy hiányossága: teljesen külön kezeli a szöveges és a képi információkat. De a fejlesztők ígérete szerint hamarosan várható, hogy azok egyidejűleg jelennek meg a képernyőn.

Gyors és biztonságos

A bevezetőben említett másik hazai fejlesztésű rendszerről (Kerestext) az Alaplapban már korábban beszámoltunk, így most csak néhány, a FrameBase-től eltérő szolgáltatására szeretnénk felhívni a figyelmet.

Talán a legfontosabb, hogy a Kerestext alkalmas a szöveghez tartozó kép (videofelvétel, alaprajz, fénykép, helyszínvázlat) megjelenítésére, akár a szöveg részeként, akár önállóan is. A felhasználó kényelmét szolgálja az a megoldás, hogy a szöveges részben kapcsos zárójelekkel különítik el a képhivatkozásokat. Ezzel a módszerrel a képhez teljes szöveg rendelhető. A Kerestext másik jellemzője, hogy az információk visszakeresése szabadon vagy (kétszintű) szótár segítségével több szempont szerint, természetes nyelven történik. A program kezeli az előjegyzéseket, vagyis az információbázisból a kapcsolódó előzményeket előkereshetjük, ki-

gyűjthetjük és azonnal felhasználhatjuk. A program határidőket is kapcsolhat a dokumentumokhoz, ezek aktualitásáról a rendszer értesíti a felhasználót. A Kerestext mindezeket a funkciókat nagyon gyorsan, hatékonyan, takarékos tárolással hajtja végre. A sebességi viszonyok illusztrálására jó példa, hogy egy 18 K-s anyagot kb. fél perc alatt rak be a program az adatbázisba.

A Kerestext kezeli a teljes magyar karakterkészletet, s alkalmas tetszőleges nyelvű szöveg feldolgozására. Biztosítja a teljes körű adatvédelmet is (belépési azonosító, jelszórendszer, beavatkozási minőség). Hálózatban (Novell alatt) is működik, ahol minden munkaállomáson a képi és a szöveges adatbázishoz a láthatóságnak megfelelően férhetünk csak hozzá. A program (single üzemmódban 74 000 Ft, hálózatban 192 000 Ft) mindazokon a területeken jól használható, ahol egyrészt rengeteg információt tárolnak, másrészt ezek biztonságos kezelése, gyors bevitele és visszakeresése létfontosságú. Nem véletlen, hogy a Miniszterelnöki Hivatal, a kormányőrség és a rendőrség máris használja. Ugyanakkor jól használható gazdasági, kereskedelmi, piaci információk gyűjtésére és feldolgozására, sajtó- és témafigyelésre, könyvtári és egyéb katalógusok kezelésére. A magyarul beszélő képi és szöveges adatbázis-kezelők elérhető áron, magas szintű szolgáltatásokkal valamennyiünk számára hozzáférhetőek. Csak el kell döntenünk, hogy melyiket „szeressük".

Sziebig Andrea



INFORMÁCIÓKÉRÉS: 11 ▲

INFORMÁCIÓKÉRÉS: 05 ▲

INFORMÁCIÓKÉRÉS: 04 ▲

Instacalc

Egy kisegér és az elefántok

A táblázatkezelők mára talán már kivívták méltán megérdemelt helyüket az alapvető fontosságú szoftverek világában.
Üzleti célú alkalmazásukon túl sok segítséget jelenthetnek olyan területeken is, ahol az adatok táblázatba szervezése és az ezek segítségével történő kiértékelés elkerülhetetlen feladat.

A felhasználó ma már több cég egymást felülmúlni akaró terméke közül választhatja ki a megoldandó feladatnak leginkább megfelelőt. E termékek mindegyike képes bonyolultabbnál bonyolultabb feladatokat elvégezni. Persze ennek ára van. A probléma elsődlegesen az említett szoftverek helyigényénél jelenkezik. „Súlyuk" több száz kilótól néhány megáig terjed, s így a tárolási helyigény/kihasználtság hányados nagyon megdrágítja a hordozható számítógépek használatát.

Emellett a menürendszerek túlontúl kusza világa lassítja a könnyed kezelhetőséget, ami éppen a gyors alkalmazásoknak szab határt. Az egyszerűbb feladatok megoldása is bonyolult időtöltéssé válhat. Az előbb említett problémákra kínál hatékony orvosságot az Instacalc néven futó program. A precíz, lényegretörő dokumentációt is tartalmazó programcsomag a SolarSoft programkönyvtár 299-es lemezén található. Nézzük, mivel rukkol ki ez a kisegér a megannyi elefánt között! A program leglényegesebb — és valljuk be, a táblázatkezelők között a legrendhagyóbb — vonása az, hogy memóriarezidens. Ez azt jelenti, hogy a program indításakor betöltődik a memóriába, s türelmesen vár arra, hogy a felhasználó az Alt-I billentyűpárral aktivizálja. A program 128 K területet lop el a RAM-ból. Ebből 44 K terület szolgál az éppen aktív tábla tárolására. Az aktiválás után, az előbb említettek hatására az aktuális munkatábla egy pillanat alatt megjelenik. Gyorsasága mellett az is előnye, hogy olyan egyedi felhasználást is lehetővé tesz, ami könnyebbé teheti a táblázatkezelőkkel nap mint nap dolgozók munkáját.

Ha a munkatábla kitöltése összekapcsolódik egy hozzá kapcsolódó szöveg szerkesztésével, például egy jelentés megírásával, nincs szükség arra, hogy a munkát megszakítva tébláboljunk a programok között. Az aktivizálás után máris a szükséges táblázatban dolgozhatunk, ezt elhagyva folytathatjuk a szövegszerkesztést.

A szoftver úgy próbál hasznos segítőtárssá válni feladatok gyors elvégzésében, hogy közben kompatibilis maradjon a nagyobb rendszerekkel. Ennek elősegítésére a programot hasznos kiegészítőkkel vértezték fel. Képes Lotus 1-2-3- és dBase III-állományokat is beolvasni, illetve az aktuális munkatáblát ezeknek megfelelő formátumban tárolni.

Rendelkezik a táblázatkezelőket jellemző alapvető funkciókkal, megtalálhatók a legfontosabb belső függvények, makródefiníciókat kreálhatunk, az adatokat vonaldiagramokban ábrázolhatjuk. Mindent az egyszerűség és a könnyű áttekinthetőség jellemez.

Ha nem kívánunk tovább dolgozni, s a rezidens állomány a memóriában zavaróvá válna, egy mellékelt segédprogrammal eltávolítható onnan. Ajánlom azoknak is ezt a programot, akik most kezdik az ismerkedést a táblázatkezelőkkel.

Szalóczi Béla

MARK/FMARK/RELEASE/MAPMEM

Takarodót fúj a rezidensnek

Ki ne találkozott volna már azzal a gonddal, hogy memóriarezidens (TSR) programjai miatt nem tudta elindítani kedvenc böhönceit (Lotus, Word stb.)? Ilyenkor az ember többnyire szép lassan hátulról visszafelé haladva kigyomlálgatja szelídebb TSR programjait, hogy dolgozni tudjon. A legtöbb esetben (mondhatni, mindig) azt tapasztaljuk, hogy néha még az Alt/Ctrl/Del sem segít, mert a billentyűzetet is sikerrel lefagyasztottuk — jöhet a reset!

Szerte a világon sok programozót bosszantottak már a fenti jelenségek, s mit ad isten, még megoldásfélét is találtak. Az egyik legelterjedtebb programcsomag a TurboPower Software nevével fémjelzett — ugyan melyik programozó nem találkozott még valamelyik termékükkel? — MARK - FMARK - RELEASE - MAPMEM csomag, amely apró TSR-kezelő utility programok gyűjteménye. A programok egyszerre több verzióban is megtalálhatók a SolarSoft Programkönyvtár 469/2-es lemezén. Emellett még egyéb SolarSoft lemezeken is megtalálhatók (SolarSoft 447—451, TSR-MIX 1—5).

Pár éve egészen véletlenül került kezembe ez a négy apró programocska. Használatuk viszonylag egyszerű, segítségükkel könnyen és gyorsan el lehet bánni az átmenetileg feleslegessé vált TSR programokkal.

1. MARK.COM

Megjelöli a memóriát, foglaltsági térképet készít. Minden egyes „markolásnál" nevet is adhatunk. A markoláshoz használt nevekről annyit, hogy akár 126 karakter hosszúak is lehetnek (más kérdés, hogy ki használ ilyen hosszú neveket). A névben nem lehet szóköz és tabulátor karakter, a kis- és nagybetűket nem különbözteti meg a program. A ! karakternek a név elején különleges szerepe van.

2. MAPMEM.EXE vagy MAP.EXE

Kijelzi a képernyőn, mi minden tanyázik rezidensen a memóriában, mennyi helyet foglal el, és a nála nem frissebb verziójú MARK vagy FMARK programmal elhelyezett jelöléseket is kiírja.

3. RELEASE.EXE vagy UNMARK.COM

Felszabadítja MARK és FMARK segítségével megjelölt helyig a RAM-ot.

4. FMARK.COM

Ugyanúgy működik, mint a MARK, azzal az apró különbséggel, hogy a RELEASE, illetve az UNMARK program működéséhez szükséges adatokat nem a RAM-ban tárolja, hanem az FMARK megadásakor megnyitott állományban. Ezáltal 1300-1500 bájt helyett (minden verzióban más a MARK rezidens mérete) csak mintegy 160 bájtot foglal le.

Alapesetben csak megjelöljük a memóriát, a TSR-ek eltávolítására a RELEASE szolgál. A foglaltsági státuszokat a TSR behívása előtt névvel jelöljük meg (SK — Sidekick, AMI — AMI stb.).

A névmegadás nélkül eleresztett RELEASE a legutolsó „markolásig" szabadítja fel a memóriát. Ha nevet is adunk a RELEASE mellett, akkor a megadott névvel végzett markolásig felszabadítja a memóriát. Védett jelölés megadására a ! karakter szolgál (MARK !KAKUKK). Felszabadítani ilyenkor csak teljesen azonos névmegadással lehet (RELEASE !KAKUKK).

Sem a RELEASE, sem a RELEASE KAKUKK nem szabadítja fel, csak közli, hogy védett a jel.

Ugyanúgy hibaüzenetet kapunk, ha az FMARK-ot használtuk a jelölésre, és a RELEASE mellett nem adtuk meg az FMARK mellett alkalmazott pontos állománynevet.

Fel kell hívnom a figyelmet néhány apróságra:

1. A markolás is helyet foglal a RAM-ból.

2. Kínos, ha elfelejtjük, milyen nevet is adtunk a MARK mellett. Ilyenkor ne szégyelljük megnézni a MAPMEM-mel. Ha az adott névvel sem takarodik ki, akkor ! jelet alkalmazzunk.

3. Kellemetlen rendszerlefagyásokhoz vezet, ha ész nélkül, rendszertelenül alkalmazzuk a MARK és RELEASE programokat. Célszerű például a SideKick meghívását egy markolással kiegészített batch fájlból intézni.

A Lotus indítását pedig a UNSK_123.BAT segítségével végezzük, ha meggyőződtünk arról, hogy a SideKick még a memóriában garázdálkodik.

4. Ha nem volt semmi TSR a RAM-ban, egy RELEASE paranccsal esetenként elegánsan le lehet merevíteni a rendszert. Ezért a RELEASE kiadása előtt hacsak lehet, adjuk ki a MAPMEM parancsot is ellenőrzésképpen.

5. A Norton Commander kedvelőinek fontos: ha az NC indítása előtt markoltunk meg valamit, és azt a RELEASE vagy UNMARK programmal eleresztjük, ne lepődjünk meg, ha a gép nem talál vissza az NC-hez. Ugyanis a RELEASE az ehhez szükséges adatokat is kirámolta. Ezért ha NC-vel dolgozunk és a MARK/FMARK/RELEASE programokat használjuk, azokat mindig az NC-ből való kilépést követően indítsuk csak el!

Természetesen az ismertetett programoknak még vannak egyéb kapcsolói is, bár szerencsére (?) nem túl sok.

(Zárójelben jegyzem meg, hogy a programok alkotói a Novell NetWare alkalmazóiról sem feledkeztek meg. A MARKNET és a RELNET nekik készült, a leírás részletesen a SolarSoft 469/2-es lemezén olvasható.)

Nagy Gábor

Tmrttt állományok kiteeríiítveeee

Napjainkban a szoftverpiac bővülésével párhuzamosan egyre inkább megnőtt az igény arra, hogy a rendelkezésre álló nagy mennyiségű programot valamilyen formában tárolni tudjuk. Ennek egyenes következménye az egyre újabb és újabb tömörítő-csomagoló programok széles körű elterjedése. Ezek a tömörítőprogramok dicséretesen gyorsan és jól dolgoznak, és igen sok szolgáltatásuk is ismeretes.

Van egy aprócska probléma, ami enyhén szólva megnehezíti a tömörítők használatát: az általuk létrehozott állományokat szinte semmiféle segédprogram nem képes felismerni, értelmezni és akár csak alacsony szinten kezelni. Persze néhányan — közöttük a mindenből „kimaradhatatlan" Peter Norton és csapata — megpróbálkoztak e feladat megoldásával, és így születhetett meg például a ZipView, a Shez vagy a Narc nevű igen hasznos segédprogram is. Ezeknek azonban igen sok hátrányuk van — vagy túlzottan drágák (Shez), vagy nem ismerik a legújabb elterjedt formátumokat (ZipView), és ha véletlenül több formátumot is képesek feldolgozni, akkor sem tudják az állományokban található fájlokat egyenként megfelelően kezelni (Narc).

Ezért döntöttem úgy, hogy kifejlesztek egy olyan segédprogramot, amelynek kivitelezésekor a fent leírt hátrányokat igyekszem elsődlegesen kiküszöbölni úgy, hogy a program még emellett se legyen túl hosszú, és mind árban, mind pedig tudásban alkalmazkodjon a kis pénzű, ám igényes magyar felhasználóréteg kívánalmaihoz.

Lássuk először a formátumokat. A legfontosabb ezek közül talán az egyre elterjedtebb ARJ csomagoló által létrehozott formátum, amely rendkívüli — még az LZH-t is meghaladó — tömörsége miatt vált rövid idő alatt igen népszerűvé. Ez azonban törvényszerűen azzal járt, hogy néhány ritka kivételtől eltekintve semmi és senki nem tudott erről semmit. Ezért tartom nagy jelentőségűnek, hogy programom képes ezt az igencsak új formátumot is kezelni. Ezenkívül természetesen a PakView dolgozik a ZIP, ARC, PAK, LZH, DWC, ICE formátumokkal is.

Fontosnak tartom megjegyezni, hogy az ARC kivételével az összes említett tömörítő által képzett SFX (önkicsomagoló) állományokkal is boldogul a PakView. Nézzük, mit tehetünk az egy adott állományban található fájlokkal. Ki lehet őket egyenként csomagolni vagy kimozgatni az eredeti állományból. Ha már nincs szükségünk az adott fájlra, akkor meg is semmisíthetjük azt. Ha nem vagyunk biztosak benne, hogy melyik fájlra van szükségünk, akkor akár bele is olvashatunk a kérdéses állományba.

Ha nem tudjuk, hogy van-e elég hely ahhoz, hogy kicsomagoljunk egy vagy több állományt, akkor információt kérhetünk bármelyik meghajtóról, és így sokkal „szemétmentesebben" dolgozhatunk.

Bármikor megváltoztathatjuk a kicsomagolás célkönyvtárát, és természetesen bármikor lecserélhetjük az aktuális kilistázott állományt is. Az aktuális állományban szereplő fájlok egyébként függőlegesen egymás alá vannak kilistázva, és mellettük megtalálhatjuk az eredeti és csomagolt méretet, az eredeti állomány keletkezési dátumát, az eredeti és csomagolt állomány méreteinek arányát százalékban. Az egymás alá kilistázott fájlok között többféleképpen is lehet mozogni: vagy a megszokott módon billentyűzettel, vagy pedig a program által támogatott egérrel. Ugyanígy ki is jelölhetünk egyszerre több állományt a kilistázottak közül, a törlést, a kicsomagolást és kimozgatást így szintén egyszerre végezhetjük.

A PakView működéséhez természetesen rendelkeznünk kell azokkal a csomagolóprogramokkal, amelyeknek az állományait kezelni akarjuk, és az állományokba való belepillantáshoz szükségünk van a Norton WPView-ra is. Az Alaplaphoz mágneslemez mellékletén megtalálható a PakView demóverziója, amely rengeteg korlátozást tartalmaz, mindezek ellenére mérhető rajta a program hasznossága.

Dr.ótos Gábor

A PakView—NcMain—WpView teljes értékű verziója (DD floppyn) 500, a PakView—Ncmain—WpView—ZIP—ARJ—LZH—DWC—PAK—ICE csomagolók (HD floppyn) 1000 forintért szerezhetők be a fejlesztőtől. (Szerkesztőségünk minden érdeklődőnek készséggel megadja a címet, ahol a programok elérhetők.)

ChkVir

Vírusirtó magyar közkincs

A vírusirtás Magyarországon úgyszólván népmozgalommá nőtte ki magát, s ennek egyik élharcosa volt — és jelenleg is az — a ChkVir nevű komplex víruskereső és vírusölő program, amely már meglévő fertőzés esetén nyújt segítséget. A program 1991 decembere óta a SolarSoft Programkönyvtár magyar szekcióját gazdagítja, lajstromba vételi száma M048.

A program lehetőségei

A ChkVir program elsődleges feladata, hogy felderítse a számítógépes vírusok által megfertőzött állományokat, területeket. A felderítés után alkalmas továbbá arra is, hogy az ismert vírusok esetén a fertőzést megszüntesse, az állományokat és más területeket az eredeti állapotba visszaállítsa. Előfordulhat az az eset is, hogy a vírusfertőzés révén az eredeti állomány nem állítható vissza tökéletesen (a vírus destruktív tulajdonsága miatt). Léteznek olyan vírusok is (például Invaders), amelyeknél elméletileg nem minden esetben dönthető el, hogy a vírus visszaállíthatóan fertőzte-e meg a programot, vagy tönkre is tette azt. Ebben az esetben a ChkVir megpróbálja a legjobb tudása szerint visszaállítani az eredeti állományt, de nem biztos, hogy ez a visszaállítás tökéletes.

A ChkVir program további feladata, hogy felderítse az olyan futtatható programokat, amelyek valamilyen tömörítő programmal lettek sűrítve. A tömörített, futtatható állományok detektálása azért fontos, mert ezek az állományok tartalmazhatnak vírust. Előfordulhat ugyanis az az eset, hogy az eredeti programot megfertőzte a vírus, majd ezután lett tömörítve. Meg kell jegyeznünk azonban, hogy eredeti, gyári programok között is találhatunk tömörítetteket.

Víruskeresési eljárások

A ChkVir több víruskeresési eljárást egyesít magában. Mindegyik víruskeresési eljárás a futtatható állományokon, illetve aktivizálható területeken (partíciós tábla, boot szektor) hajtódik végre.

Általános víruskeresés

Az általános víruskeresés segítségével vírusonként és állományonként (beleértve az aktivizálható területeket: partíciós táblát, boot szektort) egy gyorskeresés történik. A módszer csak a szerzők által ismert vírus esetén használható, ugyanis a konkrét vírus keresése csak azon a területen, az állomány azon részében történik, ahol a vírus egyáltalán előfordulhat. Ezek alapján a víruskereséssel egyidejűleg gyorsellenőrzést hajtunk végre, a ChkVir programnak nem kell végigolvasnia a gyanúsított állományt.

Szekvenciakeresés

A vírus szekvenciájának azt a jellemző bájtsorozatot nevezzük, amely a vírus minden megjelenési formájában megtalálható.

Előfordulhat, hogy a vírus fertőzésenként változtatja magát, ekkor egy véges sok elemet tartalmazó szekvenciahalmaz segítségével dolgozhatunk. A szekvenciák kellő hosszúsága révén elenyésző lehet annak a valószínűsége, hogy a szekvencia valamely nem fertőzött állományban, véletlen egyezés folytán előforduljon. Ezzel csökken a vakriasztás valószínűsége, de teljes mértékben nem zárható ki.

A szekvenciakeresés során állományonként (beleértve az aktivizálható területeket: partíciós tábla, boot szektor) egy keresés történik, olyan algoritmus alapján, amelynek segítségével egyszerre az összes szekvenciát keressük az adott állományban, területen. Az algoritmus időigényessége NEM függ a szekvenciák számától, sokkal inkább befolyásolja az olvasási műveletek lassúsága.

Mutációk keresése

Az általános keresésben említett, a szerzők által ismert vírusok mutációinak a keresése is lehetséges a ChkVir segítségével. A ChkVir tartalmaz egy beépített 8086 processzoremulátort, amelynek segítségével végre tudja hajtani az Assembly szintű gépi utasításokat. A ChkVir a gyanúsított programot elkezdi futtatni az emuláció segítségével. A rögzített memóriaterületen ellenőrzött végrehajtás történik. Nem lesznek végrehajtva az IN, OUT, INT utasítások. A végrehajtott utasításokról pedig statisztika készül, amelynek alapján megállapítható, hogy a gyanúsított fájl fertőzött-e valamely ismert vírussal vagy annak valamilyen mutánsával.

Ismeretlen vírusok keresése

Az ismeretlen vírusok keresése is a mutációkeresésnél említett 8086 processzoremulátor segítségével történik. A mutációkereséstől eltérően itt nem statisztika készül, hanem bizonyos, a vírusokra általánosan jellemző tevékenységeket detektálunk.

A ChkVir használata

A ChkVir program parancssor és a menü üzemmódjában egyaránt működik a program minden keresési lehetősége és az opcionális lehetőségek is közel azonosak.

A ChkVir programot a következők szerint indíthatjuk:

CHKVIR <Enter> Menüvezérelt rendszer, minden tevékenység interaktívan történik.

CHKVIR ? <Enter> Segítség a lehetséges opciók ismertetésével, valamint az ismert vírusok és szekvenciák felsorolásával.

CHKVIR [paraméterek] <Enter> parancssor üzemmód. A program a megadott paraméterek szerint fut le.

A program mindkét üzemmódjában először a memóriát ellenőrzi. A memóriaellenőrzés 1 Mbájtig történik a szekvenciakeresés elvén, függetlenül attól, hogy mennyi a gép fizikai memóriája. Amennyiben a program a memóriában szekvenciát találna, úgy célszerű a gé-

pet egy biztosan vírusmentes lemezről újraindítani.

Parancssor üzemmódban a ChkVir a CHKVIR [könyvtárnév] [opciók] paranccsal indítható, ahol a könyvtárnév annak az alkönyvtárnak a neve, ahol a vírusokat akarja keresni. A program az alkönyvtár alkönyvtáraiban is megvizsgálja a fájlokat.

A menü üzemmód

A ChkVir indítására a bejelentkező kép jelenik meg. Bármely billentyű vagy az egér valamely gombjának a lenyomására végrehajtódik a memória ellenőrzése. A memóriaellenőrzés után a ChkVir feltérképezi az aktuális meghajtó könyvtárszerkezetét. Ha ez megtörtént, a könyvtárszerkezet megjelenik a képernyőn, és egy pop-up menürendszerbe jutunk. A menüben való közlekedésre a nyilak használhatók, az <Enter> billentyű hatására az aktuális parancs végrehajtódik. Az egyes parancsok indítására használható még az <Alt> billentyű az adott menüpont kiemelt betűjének megfelelő gomb lenyomásával, illetve a fontosabb parancsok esetében a funkcióbillentyűk.

A képernyő közepén felrajzolt könyvtárszerkezetben a <Space> billentyűvel vagy az egér bal oldali gombjával választhatjuk ki az ellenőrzendő alkönyvtárakat.

A menüpontok adta lehetőségek

About

Az About menüpontban a program szerzőinek információja található, valamint a program verziója és sorozatszáma.

Drives

A Drives menüpont hatására megjelenik az elérhető egységek listája, amelyekből választva megtörténik a kiválasztott egység könyvtárszerkezetének a feltérképezése.

Check

A Check menüpontban indíthatjuk a víruskeresést kiölési lehetőséggel vagy anélkül. A két lehetséges indítás közül menüben választhatunk. Amennyiben a kiválasztott egység nem hálózati, úgy először

Checking boot sector of drive X:

vagy a

Checking partition tables and boot sectors of hard drives

üzenet jelenik meg aszerint, hogy a kiválasztott egység merevlemez-e vagy sem. Ezt követően, ha nem hálózati egységet ellenőrzünk, a

Checking absolute sectors of drive X: ...

üzenetet kapjuk. Ekkor a DOS fájlstruktúráját manipuláló, de fájlokhoz kötődő vírusok (például Cluster Buster) ellenőrzése folyik.

Ezután következik a kijelölt alkönyvtárakban lévő fájlok vizsgálata. A jobb felső sarokban egy kis ablak jelenik meg, ahol vizsgált, és éppen vizsgálat alatt álló fájlok láthatók. A felrajzolt könyvtárszerkezetben az aktuális alkönyvtár is folyamatosan követi a vizsgálat menetét.

Amennyiben a vizsgálat valamely pontjánál a program vírust talált, úgy eme tényt egy figyelmeztető ablak megjelenése jelzi.

Leitold Ferenc

SolarSoft sikerlista

Az 1991. szeptember—novemberi eladások alapján

No.		Db	Programnév	Programleírás
1.	510	1	ARJ 2.21	Az eddigi legszuperebb tömörítő
2.	319	1	SCAN84 & CLEAN84	MacAfee jó öreg VIRSCAN vírusvadász programja
3.	475	1	NEWSPACE	Winchesterduplázó kisebb igényűeknek
4.	509	1	ZIPVIEW/LHA2.13/LHAUTO	A legfrissebb LHARC-verzió Yoshitól
5.	512	1	STUPENDOS & PKZMENU	PKWare DOS-shell és menüvezérelt PKUNZIP
6.	513	1	ECB	Szakácskönyv-készítő program
7.	466	1	SKYGLOBE STAR GAZER	A mozgó csillagtérkép!
8.	419	2	MODEL-S	Fejlesztőrendszer CASE dBASE-hez!
9.	329	1	PC-MAGAZINE BENCH.	A PC MAGAZINE tesztprogram-gyűjteménye
10.	441	1	DATABASE IN C	Adatbázis-kezelés (Btree, dBASE) C-ben
11.	435	1	OPTIKS & ICONVERT	Grafikus konverterek
12.	470	1	MULTI-EDIT 5.0	A legjobbnak tartott editor
13.	421	1	PKZ110 & ZIPDMP & SHEZ	Az adattömörítés klasszikusa
14.	463	1	GAMES FOR WINDOWS	10+1 játék Windows 3.0 alá
15.	418	1	FAST/SOFA/FFD	Egy új gépközeli nyelv!
16.	442	1	WINDOW PRO 1.51	No.1 ablaktechnika small/med/large
17.	344	1	LOVEDOS	Ezt biztosan szeretni fogja!
18.	M023	1	KEYBDRV	Teljes magyar ékezetesítő
19.	262	1	PIANOMAN	Zeneszerzést segítő, nagyszerű zenei program
20.	423	1	QFONT 1.15b	Fonteditor magyar Venturához
21.	422	1	HP FONTEDIT 5.7	Fonteditor lézernyomtatóhoz
22.	436	1	EMS UTILITIES	EMS-kezelő segédprogramok
23.	444	1	C-MIX #2	Rezidens programok készítése
24.	M024	1	BLISS	Főkönyvi rendszer C-ben
25.	424	1	SSQL	Mini SQL interpreter
26.	440	1	CHESS	Hét sakkprogram + egy sakkóra
27.	480	1	GRASP 1.10C	Látványos animációkészítő program
28.	494	1	TEGLP WINDOWS TOOLKIT	Ikongrafikus felület, ikoneditor Pascalhoz
29.	495	1	TEGLC WINDOWS TOOLKIT	Ikongrafikus felület, ikoneditor C-hez
30.	507	1	PC-BROWSE & NG_MAKER	Hipertext és fájlböngésző + NG decompiler

A NyelvÉsz és a Lektor

Az 1991. évi IFABO-n lehetett először megvásárolni az első magyar helyesírás-ellenőrző programot, a NyelvÉszt.
Megjelenésével nem csak egy új termék jelent meg a szoftverpiacon, nem csak egy régóta hiányzó eszközt kaptak kezükbe a felhasználók.
A NyelvÉsz szerzői, Béres Tibor, Hámori Miklós, Vanczák József szoftverfejlesztő mérnökök, s jómagam, a program nyelvésze bebizonyítottuk, hogy minden kétkedés ellenére is el lehet készíteni egy „magyar spelling checker"-t.

Miben új a NyelvÉsz?

Bár a programról elég sok tájékoztató, értékelő és összehasonlító írás jelent meg a legkülönbözőbb lapokban, érdekes módon senkinek sem tűnt fel igazán, hogy a korábbi, idegen nyelvű, hasonló jellegű termékekhez képest a NyelvÉsz teljesen új koncepción alapul. Ismereteim szerint ugyanis ez az első, valamilyen formában egy nyelvet feldolgozó program, amely az ún. generatív nyelvelméletből kiindulva működik.

E működés lényege az, hogy a magyar nyelv szavait a program csak egy alakban tárolja, viszont a toldalékos (azaz a képzős, ragos, jeles) szóalakváltozatokat is tartalmazza, vagyis az alapszavak alaktani származékait maga a program generálja.

Ahhoz, hogy ez a rendszer megfelelően működjön, a magyar nyelv alaktanának olyan leírását kellett elkészíteni, amely minden egyes szó minden egyes toldalékos változatát tartalmazza. Abból indultunk ki ugyanis, hogy egy spelling checker csak akkor használható rendeltetésének megfelelően, ha csakis a helyes, a létező, „A magyar helyesírás szabályai"-val megegyező írásképet minősíti helyesnek. Ezen eljárásból két előny is fakadt. Az egyik, hogy maga a nyelvi leírás, illetve ennek egy sajátosan új jellegű számítógépes nyelvi változata maga vezérli a szavak alaktani változatainak a generálását. A másik: a NyelvÉsz futtatásához 300 kbájtnál is kevesebb operatív memória szükséges. Ennek előnyeit nem is érdemes különösebben taglalni, pusztán kettőt említek. Mivel a memóriában van, a program futása önmagában gyorsabb, a szükséges tárkapacitásnak az adatok számához viszonyított igen kis volta következtében pedig könnyen és egyszerűen összeépíthető más programokkal, például szövegszerkesztőkkel. Már 1991 nyarán a program három szövegszerkesztőbe beépítve futott, a Mc Intosban, az Ékszerben és a Unix Lurixban.

Ezzel az új koncepcióval egyébként bármilyen más nyelv leírása is elkészíthető, s olyan idegen nyelvi helyesírás-ellenőrző program szerkeszthető, amely az adott nyelven mindeddig létezőkhöz képest gyorsabb, pontosabb, s ami a legfontosabb, lényegesen kisebb terjedelmű.

Amivel nem foglalkoztunk

A szerzők a NyelvÉszt kimondottan spelling checkernek szánták, s nem kívántak foglalkozni a szavak egybe- vagy különírásának kérdéseivel. Már csak azért sem, mert a magyar nyelvben szinte végtelen azoknak a szószerkezeteknek a száma, amelyek egybe is és külön is írhatók, részint a jelentésbeli különbség (melegágy, de: meleg ágy), részint az eltérő grammatikai szerkezetek használata (mosónő, de: ruhát mosó nő) miatt.

Ha a felhasználó tudatában van annak, hogy a szavak egybeírásából fakadó hibákat a program nem jelzi, ettől függetlenül is jó hatásfokkal használhatja, legalábbis ilyen visszajelzések jutottak el hozzám. A bírálatok viszont éppen ebben látták a NyelvÉsz gyenge pontját. Előfordult ugyanis, hogy a program a nyilvánvaló helyesírási hibát összetett szóként felismerve és értelmezve helyesnek minősítette. Ez a felhasználók számára nem annyira hibaforrást, mint inkább kényelmetlenséget jelentett.

Miért készült az új program?

Okulva a piac által diktált feltételekből, a kapott visszajelzésekből, a közölt kritikákból, az Ifabo után nyomban nekiláttunk, hogy a programot a legkényesebb, a legkritikusabb igényeket is kielégítővé tegyük. Hamarosan rájöttünk azonban arra, hogy az a nyelvi leírás, amely a betűellenőrzést a szavak minden alaktani változatában lehetővé teszi, bizonyos nyelvi-nyelvtani és helyesírási kérdések kezeléséhez nem kielégítő. Ezután ki kellett alakítanunk egy olyan új programszerkezetet, amely alkalmas arra, hogy minden olyan nyelvi jelenséget, amelynek ismerete egy igényes helyesírási programtól elvárható, kielégítően kezelhessünk.

Miben régi, miben új a Lektor?

A Lektor elsősorban az alapelvében azonos korábbi programunkkal, megtartottuk ugyanis a generatív nyelvszemléletből fakadó generatív felépítési elvet, a belőle származó összes előnnyel együtt. Változatlanul ragaszkodtunk ahhoz, hogy az új program is csak a létező alaktani változatokat minősíthesse helyesnek az érvényes helyesírási szabályok alapján.

Teljesen új viszont az adatbázis és a nyelvtani leírás. Míg a NyelvÉsz esetében a feldolgozott szókészlet A magyar nyelv szóvégmutató szótárán alapult, az abban szereplő szókincset javítottuk és bővítettük, a Lektor szóadatbázisa több forrásból származik. Többek között jó néhány megabájtnyi, különböző sajtótermékekben már megjelent szövegből merítettünk. Elmondhatjuk, hogy a szóadatbázis egy része saját gyűjtésünk eredményeképpen került a programba.

A szókincs és a nyelvtan

A kibővített szókincs új elrendezésben szerepel a Lektorban. Erre azért is szükség volt, mert az új szolgáltatások a korábbinál finomabb, részletesebb felbontást igényeltek.

Az új nyelvtani leírás sokkal rövidebb. Például a NyelvÉszben szereplő főnevek alaktani generálásának leírása mintegy 190 kézzel írt oldalt tett ki, a Lektor esetében ez mindössze 30 oldalnyi, annak ellenére, hogy a leírás részletesebb, pontosabb, bizonyos nyelvtani információkat is tartalmaz. Ugyanakkor a feldolgozott főnevek száma jelentősen nagyobb. A nyelvtani leírás lehetővé teszi, hogy az ún. továbbképzett szóalakokat is helyesen kezelhessük. Hogy ez mennyire bonyolult kérdés a magyar nyelvben, arra csak egy példa: számos esetben előfordul, hogy egy szóhoz ugyanaz a képző ismételten hozzátehető (igaz-ság-os-ság). A Lektor is kezeli a szóelválasztás minden esetét, sőt az ún. „hathármas" szabályt is „ismeri".

A legnagyobb újdonság azonban az, hogy a szóösszetétel azon eseteit, amelyek matematikailag definiálhatók, s amelyek nem kezelése komoly helyesírási hibát eredményez, igyekeztünk minél nagyobb számban feltárni és kezelni. Például a Lektor nem engedi meg, hogy valaki véletlenül is egybeírja a „nem" módosítószót bármelyik főnévvel. Nincs tehát „nemasztal", de van „nemlineáris" szavunk.

Valójában a Lektor egy sajátos, a feladatainak megfelelő nyelvi elemző programrészt is tartalmaz. Ennek egyik érdekessége, hogy csekély munkával átalakítható sajátos nyelvészeti kutatások céljaira is. Így például az átalakított programot használva néhány heti munkával előállítható a ma létezőkkel egyenértékű gyakorisági szótár.

A szókincs növekedésének és a pontosabb, a lehetőség szerinti legrészletesebb nyelvtani leírásnak az lett az eredménye, hogy új programunk mintegy 25 milliárd szóalakváltozatot ismer — az igekötős alakokkal együtt, de az összetett szavak nélkül.

Mindez kevesebb mint 300 kbájton. Sebességéről pedig csak annyit, hogy egy 4 oldalas könyvnek megfelelő szöveg hibalistájának az elkészíttetése a leglassabb XT gépeken sem több 1 percnél. Egy Apple McIntosh gépen egy normál oldalnyi gépelt szöveg ellenőrzése a hibák számától függően mintegy 1 másodpercig tart.

A Lektor rövidesen minden, Magyarországon általában használatos szövegszerkesztővel egybeépítve is megvásárolható.

Seregy Lajos

netben a címkék és ugróutasítások lokalizálása és legalizálása történik. Második menetben az utasításokat ún. belső kódra fordítja és a címkéket behelyettesíti a nekik megfelelő sorszámmal. Parancskészlete bővebb, mint a Norton Utilities Batch Enhancer programé. A program dokumentációja magyar nyelven található meg a lemezen.

Lemezszám: M023

Név: KEYBDRV

Szerző: Varga Balázs

Leírás: Rugalmas többfunkciós billentyűzetmeghajtó program, teljes magyar ékezetesítő. A billentyűzet váltásával szinkronban váltja az EGA monitor és a printer betűképeit. A szoftver bármikor gyors helpet ad a képernyőre. A KEYBDRVC fordító segítségével minden funkciója egy szövegfájl elkészítésével programozható. A meghajtó rendelkezésre áll COM és DOS eszközmeghajtó program formában is. A megszakításkezelő az Alt+Szóköz együttes leütésére, 80x25-ös text üzemmódban gyors help képernyőt jelenít meg, s az mindaddig látható, amíg az előbbi gombokat lenyomva tartják.

Lemezszám: M024

Név: Bliss

Szerző: Balázs László

Leírás: Korlátozásoktól mentes, szabadon kipróbálható főkönyvi könyvelési rendszer, C-ben programozva. A lemezen az FK-SETUP.EXE gondoskodik a főkönyvi könyvelést végző programcsomag telepítéséről. Maga a program a #421-es lemezen található PKZIP-pel készült önkibontó FOKONY12.EXE állományban van.
A Cédrus Kiadó azt tervezi, hogy az 1992-ben indítandó Alaplap Füzetek sorozatban megjelenteti a program kézikönyvét, mágneslemez melléklettel.

Lemezszám: M025

Név: Lexicon v.2.0

Szerző: Czifra Sándor

Leírás: Szakfolyóiratok témafigyelése. A Lexicon program alkalmas folyóiratokban megjelent publikációknak, illetve azok kivonatainak strukturált keresési szempontok szerinti nyilvántartására, folyamatos karbantartására, különböző szempontok szerinti lekérdezésére és megjelenítésére.
A program közel egymillió dokumentum felvételére képes. De nincs akadálya ennél több dokumentum nyilvántartására alkalmas program installálásának sem. A dokumentumok sorszámozása automatikusan történik.
Egy publikáció tartalmából csak a könyvtári nyilvántartás szempontjából fontos adatokat szükséges nyilvántartani, ami lényegesen kisebb terjedelmű. A felvett adatok a tárolási és a keresési szempontoknak is eleget tesznek. A dokumentumok sorszám szerint egyediek.

A Lexicon program funkciói:
— Dokumentumok felvétele.
— Dokumentumok módosítása.
— Lekérdezési, listázási lehetőségek.
— Segédrutinok.
— Szöveges fájlok olvasása.
— Törlések visszaállítása.
— Filter állomány generálása.
— Adatbázis szervezése.
— DOS parancsok, programok.
— Állapotinformációk lekérése.
— Helptámogatás adatbevitelhez.
A Lexicon program alkalmas:
— Szakmai szempontok szerinti szótárak kialakítására.
— Szakirodalom tartalmi részeinek gyors és rövid nyilvántartására.
— Különböző szempontok szerinti rugalmas és gyors lekérdezésre.
— Nyitott karbantartási funkciókra.
— Strukturált könyvtári nyilvántartásra.

Lemezszám: M026

Név: Letmark

Szerző: Czifra Sándor

Leírás: Postaforgalom iktatása. A Letmark program alkalmas levelek, levélnek minősített feljegyzések dokumentumainak, kivonatainak strukturált keresési szempontoknak megfelelő nyilvántartására, folyamatos karbantartására, különböző szempontok szerinti lekérdezésére és megjelenítésére.
A Letmark nyilvántartási funkciója fontos helyet foglal el a levelezés információs folyamatában. A levelet bármelyik szövegszerkesztővel megírhatjuk, majd megfelelő formátumban kilistázhatjuk. A levélfájlok tartalmából csak a gyors és korrekt ügyintézés szempontjából fontos adatokat szükséges nyilvántartani, ami lényegesen kevesebb, mint a levél teljes terjedelme.
A Letmark program funkciói hasonlóak a Lexicon programéhoz (#M025). Alkalmazásával megoldhatjuk a levelezés tartalmi részének gyors és rövid nyilvántartását, a rugalmas és gyors lekérdezést.

Lemezszám: M027

Név: Szociometria

Szerző: Csiki András

Leírás: A szociológiai, kisközösségi kapcsolattartást szolgáló program CGA képernyőre készült. Hercules kártya esetén CGA emulátort kell használni. EGA és VGA esetén CGA módba kell kapcsolni. A lemezen a program teljes forráskódja megtalálható. A programban bármely rész átírható, belőle rutinok kiemelhetők (public domain).
A program segít feltárni a kisközösségek tagjainak kapcsolat-

ALAPLAP KÖNYVEK

MEGRENDELEM

postai utánvétellel az ALAPLAP KÖNYVEK sorozatban megjelenő alábbi könyv(ek)et:

Nagy Gábor: Tömör gyönyör
...... példányban, á 256,– Ft

Kis János — Szegedi Imre: Új víruslélektan
...... példányban, á 256,– Ft

Kis János — Szegedi Imre: Vírushatározó
...... példányban, á 256,– Ft

Számítástechnikai alaplexikon I.
...... példányban, á 256,– Ft

A vételárat és a kézbesítési díjat a küldemény átvételekor fizetem ki. Kérem, hogy a könyv(ek)et megjelenés után az alábbi címre postázzák:

Név: ....

(Cég:) ....

Utca, házszám: ....

Helység, ir.-szám: ....

Dátum:

....
(aláírás)

APRÓHIRDETÉSI MEGRENDELŐLAP

Kérem, hogy az Alaplap következő számának Mikrobazár rovatában közöljék az alábbi szövegű apróhirdetést:

....

....

....

....

....

(Maximális terjedelem: 300 betűhely.)

PC Turbo Klub

Ezennel belépek a PC Turbo Klub tagjainak sorába. Az egy évre szóló tagsági díjat befizettem, és mellékelem az igazolószelvény másolatát.
A tagsággal járó Alaplapot és egyéb küldeményeket az alábbi címre kérem:

Név: ....

(Intézmény:) ....

Utca, házszám: ....

Helység: ....

Irányítószám: ....

1992. hó nap

....
(aláírás)

FELADÓ

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon: ....

❑ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes közlését.

❑ A hirdetés kereskedelmi célú.

Mellékelem a soronként (60 betűhelyenként) 300 forintnak megfelelő összeg átutalását igazoló bizonylat másolatát. (Címzett: Cédrus Informatikai Rt. 1251 Budapest XI., Karolina út 17. Bankszámlára: Budapest Bank 380-66760 számlaszám.)

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 71.
Budapest
1251

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 71.
Budapest
1251

Belföldön díjmentesen feladható

Cédrus Kiadó
Pf. 71.
Budapest
1251

ALAPLAP

szövegíráj... mozható. A meghajtó rendelkezésre áll COM és DOS eszközmeghajtó program formában is. A megszakításkezelő az Alt+Szóköz együttes leütésére, 80x25-ös text üzemmódban gyors help képernyőt jelenít meg, s az mindaddig látható, amíg az előbbi gombokat lenyomva tartják.

Lemezszám: **M024**

Név: **Bliss**

Szerző: Balázs László

Leírás: Korlátozásoktól mentes, szabadon kipróbálható főkönyvi könyvelési rendszer, C-ben programozva. A lemezen az FK-SETUP.EXE gondoskodik a főkönyvi könyvelést végző programcsomag telepítéséről. Maga a program a #421-es lemezen található PKZIP-pel készült önkibontó FOKONY12.EXE állományban van.
A Cédrus Kiadó azt tervezi, hogy az 1992-ben indítandó Alaplap Füzetek sorozatban megjelenteti

Egy publikáció ...
a könyvtári nyilv
pontjából fontos
ges nyilvántarta
sen kisebb terje
adatok a tárolás
szempontoknak
nek. A dokumen
szerint egyediek

A Lexicon prog
— Dokumentum
— Dokumentum
— Lekérdezési, ségek.
— Segédrutinok
— Szöveges fájl
— Törlések viss
— Filter állomán
— Adatbázis sze
— DOS parancs
— Állapotinform
— Helptámogat
A Lexicon progr
— Szakmai sze szótárak kialakí
— Szakirodalon nek gyors és rö ra.
— Különböző s ti rugalmas és g re.
— Nyitott karba ra.

A LEMEZMELLÉKLET TARTALMA:

- Adventi kalendárium (Anagramm, Chkmemo, Filefind)
- Beszállókártya a DOS-hoz (Demó változat)
- Egy csomagdemó a tömörítőkhöz
- Az Alaplap eddigi számainak tematikus tartalomjegyzéke
- Játék: Ugrabugra



Nekünk a biztonság a fontos.

Mi Polaroid mágneslemezt használunk.

Minden célra a megfelelő minőséget!

A Polaroid számítástechnikai termékeinek kibővített választéka

MÁGNESLEMEZEK

DataRescue

A csúcstechnológia maximális adatbiztonságú terméke

Cikkszám	Típus	Doboz	Ára*/10 db
607339	5,25" DS/DD, 360 kbájt	Karton	900,–
607348	5,25" DS/DD, 360 kbájt	Dial'N'File	1 200,–
608622	5,25" DS/HD, 1,2 Mbájt	Karton	1 700,–
611465	5,25" DS/HD, 1,2 Mbájt	Dial'N'File	2 000,–
608623	3,5" MF/2DD, 720 kbájt	Karton	1 800,–
615655	3,5" MF/2DD, 720 kbájt	Dial'N'File	2 100,–
612394	3,5" MF/2HD, 1,44 Mbájt	Karton	2 700,–
615656	3,5" MF/2HD, 1,44 Mbájt	Dial'N'File	3 000,–
607344	8" DS/DD, 1,6 Mbájt	Karton	3 000,–

Professional Quality

Megbízható minőség a hivatásos felhasználóknak

Cikkszám	Típus	Doboz	Ára*/10 db
614456	5,25" DS/DD, 360 kbájt	Karton	660,–
610730	5,25" DS/DD, 360 kbájt	Műanyag	860,–
614455	5,25" DS/HD, 1,2 Mbájt	Karton	1 100,–
610732	5,25" DS/HD, 1,2 Mbájt	Műanyag	1 300,–
614454	3,5" MF/2DD, 720 kbájt	Karton	1 200,–
610733	3,5" MF/2DD, 720 kbájt	Műanyag	1 400,–
614453	3,5" MF/2HD, 1,44 Mbájt	Karton	2 200,–
610734	3,5" MF/2HD, 1,44 Mbájt	Műanyag	2 400,–

STREAMER KAZETTÁK

Cikkszám	Típus	Ára /db
610725	300 XL/P, 45 Mbájt	2 700,–
610729	600 XT/P, 300 Mbájt	3 000,–
612544	2000/P, 40 Mbájt	2 150,–
612545	600/P, 120 Mbájt	2 900,–

KÖRPOLARIZÁTOROS MONITORSZŰRŐK

Cikkszám	Típus	Ára*/db
615001	CP Universal II, 12"	8 500,–
615002	CP Universal II, 14"	8 500,–
614021	CP Universal, 12"	18 800,–
614020	CP Universal, 14"	18 800,–
614956	CP Workstation, 19-21"	26 800,–

TISZTÍTÓK

PolaClear tisztítókészlet 900,–

**Az árak az áfát nem tartalmazzák.*

FLOPPYLAND • Budapest V., Váci utca 84. • Telefon/Telefax: 118–2651

Cédrus Karolina Áruház • Budapest XI., Karolina út 17. • Telefon: 185–2421/152

Árusítás még a Polaroid számítástechnikai termékeinek jogosított viszonteladóinál is:

BUDAPEST

Bit & S Kft.
II., Tárogató u. 2–4.
Tel.: 176-2778

Cash Bt.
V., Október u. 16.
Tel.: 112-3764

Cobra Ksz.
VII., Király u. 9.
Tel.: 142-2740

Controll Rt.
IX., Üllői út 101.
Tel.: 114-0211

Galax Kft.
XI., Bocskai út 54.
Tel.: 166-7557

Microsystem
XII., Városmajor u. 74.
Tel.: 156-5366

Mikroszerviz Kft.
IV., Templom u. 7.
Tel.: 189-0272

Omikron Ksz.
XI., Bartók Béla út 134.
Tel.: 186-9967

PC-Pince
VI., Nagymező u. 64.
Tel.: 132-7751

Pixel G. Kft.
XI., Fadrusz u. 23.
Tel.: 165-2805

Polaroid Mintabolt
VII., Klauzál tér 11.
Tel.: 142-2733

Selectrade
II., Mihályfi út 29.
Tel.: 176-4800

Sincord Kft.
Rákóczi út 4–6.
(Arissimeone Center, fsz.)
Tel.: 142-3542/24

Softinvest
V., Jászai Mari tér 3.
Tel.: 112-4873

Swisscad Kft.
XII., Márvány u. 23.
Tel.: 155-0393

Számalk Interfész
I., Donáti u. 44.
Tel.: 115-8090

Számalk Szaküzlet
XI., Szakasits út 68.
Tel.: 185-1294

Szerval Kft.
VI., Szófia u. 5.
Tel.: 142-6938

SZÜV Computer-M Kft.
VII., Wesselényi u. 21.
Tel.: 111-7917

BAJA
Computer-Market Kft.
Béke tér 7.
Tel.: (79) 22-974

BALMAZÚJVÁROS
Dignitas Kft.
Veres Péter út 1.
Tel.: (52) 70-404

BÉKÉS
Bé-Ta Kft.
Kert u. 1/1.
Tel.: (66) 41-745

BUDAÖRS
B 30 Betéti Társulás
Liliom u. 17.
Tel.: (18) 26-093

DEBRECEN
Főnixcomp Ksz.
Sumen u. 24-26.
Tel.: (52) 10-571

Inex Kft.
Hunyadi u. 13.
Tel.: (52) 11-031

DOROG
Printer Fair Kft.
Május 1. út 11/a.
Tel.: (33) 31-037

DUNAÚJVÁROS
Duna-Soft Informatikai Kft.
Városháza tér 1.
Tel.: (25) 12-211/193

EGER
Apex Gmk
Széchenyi út 22.
Tel.: (36) 13-945

Trader Kft.
Bródy Sándor u. 5.
Tel.: (36) 25-006

GYÖNGYÖS
Abacus Kft.
Beloiannisz u. 8/1.
Tel.: (37) 13-489

Megapex Gmk.
Szent Bertalan u. 2.
Tel.: (37) 11-126

KAPOSVÁR
Microcenter Kft.
Ady Endre u. 7.
Tel.: (82) 11-442

KECSKEMÉT
Polyware Kft.
Széchenyi sétány 6.
Tel.: (76) 47-117

KESZTHELY
Ramorg
Széchenyi u. 7.
Tel.: (92) 74-632

MÁTÉSZALKA
Szalka Elektronik Kft.
Felszabadulás útja 19.
Tel.: (44) 12-532

MISKOLC
Server Kft.
Toronyalja u. 13.
Tel.: (46) 47-297

NAGYKANIZSA
Ramorg
Dél-Zalai Áruház I. em.
Tel.: (92) 73-123

NYÍREGYHÁZA
Navigátor Kft.
Tünde u. 2.
Tel.: (42) 13-311

Szerval Kft.
Derkovits u. 106.
Tel.: (42) 14-450

PÉCS
M and M Pécs Kft.
Sallai u. 32.
Tel.: (72) 26-186

Online Gmk.
Bálicsi u. 11.
Tel.: (72) 22-002

PILISBOROSJENŐ
Polyform Kft.
Petőfi u. 3.
Tel.: (20) 14-021

SALGÓTARJÁN
Szervezési és Számítástechnikai Intézet
Rákóczi út 192.
Tel.: (32) 10-295

SÁTORALJAÚJHELY
Doka-Soft Kft.
Kazinczy u. 104.
Tel.: (41) 22-861

SZEGED
Fényképész Ksz.
Kárász u. 7.
Tel.: (62) 12-469

GidaSoft
Petőfi sugárút 17.
Tel.: (62) 53-238

SZEKSZÁRD
Saturnus Kft.
Béla tér 6.
Tel.: (74) 12-928

SZÉKESFEHÉRVÁR
Patelco Kft.
Nagybányai út 37.
Tel.: (22) 11-559

SZOLNOK
Holo-Comp Kft.
Dobó István u. 25.
Tel.: (56) 30-299

TATABÁNYA
Villvakisz Ksz.
Tóth-Bucsoki u. 3.
Tel.: (34) 16-144

VESZPRÉM
Expertus Kft.
Bajcsy-Zs. u. 18-20.
Tel.: (80) 20-611

ZALAEGERSZEG
Ramorg
Kosztolányi út 41.
Tel.: (92) 13-856

rendszerét. A csoport tagjainak 3 alapkérdésre kell válaszolniuk, hogy kit választanának barátnak, munkatársnak és vezetőnek. A kérdéskör természetesen tovább is bővíthető. A program elkészíti a különböző kapcsolatok mátrixait, illetve az összesített mátrixot. Az összesített táblázatban a villogó X a kiközösített, elszigetelt személyeket jelöli. (A kölcsönös felismerési hányados 0). A RAJZ_SZM program a mátrixok grafikus megjelenítését biztosítja. A személyeket népszerűségüktől függően különböző nagyságú körök jelentik. A köröket eredetileg egy koncentrikus köröket tartalmazó táblán kell elhelyezni úgy, hogy a legbelső kör tartalmazza a legtöbb kölcsönös kapcsolattal rendelkezőket.

Lemezszám: **M028**

Név: **Sura**

Szerző: Székely Béla

Leírás: Univerzális rajzolóprogram. IBM PC XT/AT személyi számítógépekre kifejlesztett rajzoló rendszer, amely a Basic, Turbo Pascal, Turbo C és Quick C felhasználóit segítheti a képernyőgrafikai munkákban.
Konfiguráció: CGA/EGA/VGA grafikus adapter kell hozzá.

Lemezszám: **M029**

Név: **OOPlus**

Szerző: Németh Viktor

Leírás: Adatbázis-kezelés Turbo Pascal 5.5-ben. A lemezen levő DBM (Data Base Manager) program elkészíti a megfelelő adatbáziskezelő program Pascal nyelvi forrásszövegét. Telefonkönyvkezelő példaprogram, az objektumorientált rész Pascal unitokban. Az OOPlus unit a Turbo Pascal objektumorientált programozást támogató 5.5 verziójához készült. Olyan objektumokat tartalmaz, melyek fájlműveletek elvégzéséhez szükségesek, és indexelt adatbázisok kezelését teszik lehetővé. Ennek megfelelően a unit tartalmazza az adat- és indexobjektumok őstípusait, amelyekből tetszőleges számú és tulajdonságú további objektum származtatható.

Lemezszám: **M030**

Név: **CAB**

Szerző: Szolek András, Gellért Andor

Leírás: A Bankár című mágneslemez újságban megjelent sorozat gyűjteményes kiadása. (CAB = computer aided banking.) A szerzők a pénzügyi matematika táblázatkezelőkkel való támogatását ismertetik .WK1 (Lotus 1-2-3) formátumában is mellékelt állományokon.

Lemezszám: **M031**

Név: **TPUDISASM**

Szerző: Csiki András

Leírás: A Turbo Pascal 6.0 megjelenése óta is sok 5.0 alatt fordított rutin cserél gazdát TPU (Turbo Pascal unit) formában. Az általános probléma a forráskód hiánya, amely nem szerezhető meg. A TP 5.0 unit teljesen más, mint az 5.5 (6.0) részére fordított unit formátum. Sok programozó emiatt nem tér át a többet tudó verzióra. A unitok visszafordítása forrásszintű Pascal programmá sajnos nem lehetséges, de létezik a TPU2ASM (disassembler) amely Assembybe visszaforgatja a programot. Az így keletkezett program a TASM segítségével újra lefordítható, s az OBJ program bármilyen környezetbe beszerkeszthető. A UNIT—OBJ konverzió a Pascal egyik hiányosságát is kiküszöböli, ugyanis az OBJ állomány linkelhető C, Modula, és más magasszintű programnyelvekbe. A rutinok paraméterezéseit sajnos nekünk kell kitalálni. A 6.0 Pascal már ismeri az ASM típust is, így az INLINE helyett közvetlenül Assemblyt is írhatunk (nincs szüség a TASM-ra). A fordítás menetéhez szükség van az 5.0 Pascal rendszer unitjaira, amelyek a TURBO.TPL fájlban találhatók. A fordítás idejére a TURBO50.TPL-t át kell nevezni TURBO.TPL-re. (A Turbo Pascal 6.0 fordító sajnos szintén nem érti a korábbi verziókkal fordított unitokat.)
Turbo Designer — Menügenerátor.
Quick Screen utility — Gyors képernyőkezelő (ASM) rutinok.
Multitasking — Egy időben több program fut.
Tlite — Kommunikációs vonal kezelése.
A fent említett shareware programok a SolarSoft könyvtárban megtalálhatók. A programok rutinjaikat TP 5.0-val lefordított unitokban tárolják. A TPU2ASM program segítségével ezen programok minden további nélkül átmenthetők a magasabb verziószámú fordítók alá.
A mintapélda egy egyszerű, de roppant hatékony egérkezelőt tartalmaz, forráskódban, Turbo Pascal 5.0-val lefordítva. A visszaforgatást a MAKE_CIN.BAT program végzi. A visszafordított ASM tartalmaz különböző ellenőrző (call) hívásokat, amelyeket a Pascal illesztett a kódba. Stack, heap, stb. ellenőrzése. Ezeket a hívásokat ki lehet hagyni. A unit végén elhelyezkedő procedúrahívást az ASM kódból el kell távolítani, a hívás helye az újra kialakított unit végén van.
A lemez tartalmazza az eredeti forráskódú, valamint az 5.0, 5.5, 6.0 unitokat.

Lemezszám: **M032**

Név: **Vám 91**

Szerző: Numerika

Leírás: 1 darab HD lemezen található a Vám 91 űrlapkitöltő program. A program segítségével kitölthetjük a 070 számú VÁM91 űrlapot. A kitöltéshez rendelkezésünkre áll egy naptár, egy zseb-

számológép és a Help is. Szinte minden a képernyőn van.
A programot az Alaplap 1991. májusi száma részletesen ismertette.

Lemezszám: M033

Név: PassivDemo

Szerző: Zombori Ferenc

Leírás: A német passzív igeragozás gyakorlása. Játékos módszer, amellyel szórakoztató formában feleleveníthetők és bővíthetők a korábban szerzett ismeretek. A feladatok megoldásához legalább alapfokú ismeretek szükségesek, de azoknak is hasznos, akik magasabb szintű németnyelv-tudással rendelkeznek.
A feladatok a passzív igeragozás gyakoroltatására készültek, de a mondatok más típusú nyelvtani „buktatókat" is tartalmazhatnak.
A program az egyes feladatok megoldását közvetlenül értékeli, majd az összes feladat megoldása után összefoglaló értékelést ad a gyakorló nyelvtudásáról.

Lemezszám: M034

Név: HelpEdit

Szerző: Bánhidi Gábor

Leírás: Helpkészítő Clipper programokhoz. Segítségével on-line helpeket lehet egyszerűen és hatékonyan szerkeszteni. A program működése a feldolgozási folyamatot követve:

1. Külső ciklus.

— Az információs ablak koordinátáinak rögzítése. Ezen keresztül történik az adatok kiírása. Az értékeket meg kell jegyeznünk, mert a programban is ezeket a koordinátákat fogjuk használni.
— Állományunk nevének rögzítése. (Ez lesz a helpállomány.) Ha még nincs begépelt állomány, létrehozzuk azt. Amennyiben nem adjuk meg az állomány kiterjesztését, a program hozzáfűzi azt. Ez után történik az állomány megnyitása.

2. Belső ciklus.

A változó nevének rögzítése, olvasása. Amennyiben nincs változó, létrehozzuk azt. Ezután elkezdődik az információ rögzítése.

3. Programunkban a következő eljárást kell alkalmaznunk.

— Paraméterek felsorolása.
— Változók értékadása.
— Állományvizsgálat. Amennyiben nem a help állomány van nyitva, úgy az előző állományt lezárjuk, majd a help után visszaállítjuk.
— Megkeressük az adott változó nevét, ha megvan, kiírathatjuk az információt.
— Eredeti állapot visszaállítása.
A program továbbfejlesztett változata tartalmazni fogja a koordináták átvételét, az indexelt állományban történő keresést.

Lemezszám: M035

Név: TDBU (Turbo DBU)

Szerző: Szalay Tamás

Leírás: A közismert DBU adatbázis-kezelő program továbbfejlesztett változata. Aki nem ismeri a DBU-t, annak pedig egy általános célú, Clipper adatbázisokat kezelő, áttekintő és listázó program. A TDBU mindazt tudja, amit a DBU, azaz létrehoz, megnyit, módosít és indexel tetszőleges Clipper adatbázisokat, a szükséges peremfeltételekkel.
A TDBU olyasmit is tud, amit a DBU nem:
— Megadhatók kvázi visszacsatolt relációk, definiálható az adatellenőrzés a táblázatban (táblázat a táblázatban).
— Bővített, jó áttekintő és listázó (browse).
— Barátságos keresési módok (megszakítható locate interaktív paraméter beolvasásával).
— Mezőnevek egyszerű keverése, fájloktól függetlenül, a fájlstruktúrák változtatása nélkül.
— Indexkulcsok gyors cseréje a megadott relációk veszélyeztetése nélkül.
— Törölt rekordok látható-rejtett kapcsolója.
— Aktuális rekord áttekintése, módosítása akár táblázatos akár oszlopos formában.
— Aktuális rekord kinyomtatása (teljesen, részlegesen) a mezők keverésével, egyetlen gombnyomásra.
— Nyomtatási modul extra igények kielégítésére is. Teljes menüvezérlés nyomtatóra és fájlba, a relácikókal kapcsolt fájlok hierarchiáját követve. Automatikus optimális fejlécgenerátor, módosítható fejlécek, valamint listakészítés 5 megadható összegfokozat segítségével a választott mezőkből.
— Deklarálható műveleti mezők (mint egy számolótáblában).
— Göngyölített mezők.
— A deklarált listák mentése és töltése a munka könnyítése végett.
— A program hálózatban is alkalmazható.

Lemezszám: M036

Név: Filecatalogus v.1.0

Szerző: Séra László

Leírás: Ez a lemezkatalogizáló program azok számára készült, akik nem tudják megjegyezni, hogy többszáz lemezükön melyik program hol található. A könnyedén létrehozott adatbázissal nyilvántarthatjuk egész lemezállományunkat (beleértve a winchester tartalmát is). Bármikor kikereshetünk adatokat az adatbázisból, és azokat tetszés szerint ki is nyomtathatjuk. A már meglévő adathalmazt természetesen bármikor változtathatjuk. A program előnye gyorsassága és rövidsége. Mellékelve van hozzá a Turbo Pascal (v.5.5) forrásnyelvű program is, így magát a programot is módosíthatjuk, átalakíthatjuk!

Sorozatunk a betűsorozatokról III.

Calamus(zi) macska egeret fog

A DTP rendszerek minősítésénél nem csupán a rendszer tudását, hanem a hozzá tartozó utility programok mennyiségét és minőségét is figyelembe kell venni. A kiadványszerkesztő programoknak együtt kell működniük a pixelgrafikus és a vektorgrafikus rajzprogramokkal, a retusprogramokkal, különféle szövegszerkesztőkkel.

A fentiek miatt nagyon fontos, hogy az egyes programok értelmezni tudják a többi program fájlformátumát, és azt be tudják olvasni a memóriába.

A kompakt rendszer programjai

Az Atari ST-n futó kiadványszerkesztő rendszerek közül a Calamus DTP kompakt rendszert alkot az Arabesqe rajzprogrammal, az Avant Vektor és a Didot Line Art vektorizáló programokkal, a DMC Fonteditorral, a Type Art fonteditorral és az Out Line Art grafikai tervezővel, a PKS Write és a 1st Word Plus szövegszerkesztőkkel.

Karakterkészletek

A Calamus alaprendszerhez kétfajta betűcsaládot ad a DMC szoftverház, a Times és a Swiss betűkészletet. A betűcsalád a 25, az 50 és a 100-as vastagságú karakterkészletet tartalmazza álló és dőlt betűtípusból. A többi attribútumot a Calamusban állíthatjuk be, nem kell hozzá új karakterkészlet. Ez a kiadványszerkesztő a Venturával és többi vetélytársával ellentétben azonos karakterkészletet használ a különböző méretű betűk megjelenítésére. A fontok nagyítását, kicsinyítését egy, a programba beépített algoritmus végzi, ezért nem kell minden nagyságban új fontkészletet generálni. A fontkészletek ráadásul nyomtatófüggetlenek, ezért egy karakterkészlettel dolgozhatunk 9 tűs mátrixnyomtatón és akár lézerlevilágítón is.
Ha ugyanolyan konfigurációban szeretnénk dolgozni Venturán, mint a Calamus DTP-n, akkor a letárolandó fontok miatt a Ventura sokkal nagyobb winchesterkapacitást igényel.

Sokakban felvetődhet, hogy ha a fontokkal kapcsolatos sokféle attribútumművelet programból beállítható a Calamusban, akkor miért pont a dőlt betűket kell külön karakterkészletben tárolni. Ez elsődlegesen azért van, mert a legtöbbet használt betűtípusok esetén más a topológiájuk az álló és a dőlt betűknek. Nem csupán a betűk megdöntését kellene megoldani, hanem azok formájának átalakítását is. Igaz, ez nem vonatkozik a talpatlan betűcsaládokra, ahol tényleg csak a betűket kell megdönteni (általában 12 fokkal). A Calamus színes változatába, az SL-be ezért már ezt a funkciót is beépítették. Ha a felhasználó talpatlan betűt használ, akkor nem kell megvásárolni vagy megtervezni a dőlt betűkészletet, hanem azt a programból megdöntheti a kívánt mértékben.

Nyomtatás

A Calamus a nyomtatás előtt a vektorfontokkal írt szövegből, az ábrákból, raszterekből, vonalakból egy grafikus ábrát állít elő a számítógép memóriájában, amit grafikus adatok formájában küld ki a megfelelő nyomtatómeghajtó segítségével a printernek, levilágítónak. Mivel az adatokat grafikus formában állítja elő, ezért lézernyomtató használata esetén nem szükséges az, hogy a nyomtatóban PostScript értelmező is legyen, ami feleslegesen drágítaná meg a nyomtató árát.

Szövegszerkesztők

A Calamus 1.09 változata a PKS Write, a 1st Word, a Microsoft Write, a Word Perfect szövegszerkesztőkkel írt szöveget attribútumhelyesen olvassa be. Más szövegszerkesztők használata esetén csak ASCII-kódban tudja beolvasni az adatokat, ami a szöveg attribútumainak elvesztését eredményezi. Az attribútumhelyes beolvasás azt jelenti, hogy ha valamelyik szövegszerkesztőben dőlt vagy vastag betűvel írtunk be szövegrészt, vagy alsó és felső indexeket használtunk, vagy aláhúztunk betűket, akkor ezeket ugyanilyen formában láthatjuk beolvasva a kiadványszerkesztőbe is. Az egyes attribútumok cseréjére is van lehetőség. A Calamus program a szöveg beolvasásakor megkérdezi, hogy az egyes attribútumokat nem akarjuk-e másik fontkészlettel helyettesíteni. Ha igen, akkor ki kell választani a megfelelőt, és ezután már azzal töltődik be a szöveg a kiadványszerkesztőbe. A szövegszerkesztők általában csupán egyfajta betűtípust használnak, az pedig, hogy a nyomtatásban melyik lesz, a nyomtatón múlik.

A Calamusnak ezzel a helyettesítő funkciójával elérhető, hogy a szövegszerkesztőben például vastag (bold) attribútummal megadott szöveg talpas betűből álljon, a normál szöveg pedig talpatlanból.

Rajzprogramok

A Calamus 1.09 a pixelgrafikát és a vektorgrafikát egyaránt támogatja. A pixelgrafikus képek közül az IMG és a CRG (Calamus Raster Graphic) fájlokat, a vektorgrafikus képek közül a GEM és a CVG (Calamus Vektor Grafika) formátumot tudja importálni és exportálni. Az SL változatot olyan vektorgrafikai feldolgozó programmodullal is kiegészítették — a speed linenal —, amely nagyon gyorsan tud pixelgrafikát vektorizálni. A programozók viszont ügyeltek arra, hogy az így kapott vektorgrafikát a program ne tudja CVG formátumban eltárolni, csak egy új, ún. CVD (Calamus Vektor Dokument)-ben. Valószínű, hogy azért módosítottak a fájlformátumon, mert a Calamus 1.09-es változathoz nagyon sok szoftverház írt különféle segédprogramokat, amelyek feldolgozzák a CVG, CRG típusú fájlokat. Az új forma bevezetésével ezek használata nehézkessé válik.

Kovács P. Attila

BASF video- és audiokazetták

BASF videokazetták		**BASF** audiokazetták	
BASF HS 200	399,–	BASF FE 60	99,–
BASF HS 240	480,–	BASF FE 60	119,–
BASF EG 200	470,–	BASF CE 60	190,–
BASF ES 240	540,–	BASF CE 90	220,–
		BASF CM 60	250,–
		BASF CM 90	285,–

A fenti árak 25% ÁFA-t tartalmaznak!

A komplett **BASF** video- és audiokazetta-árlistánkat kívánságára rendelkezésére bocsátjuk.

Az

UNITRADE
Szervezési, kereskedelmi és Számítástechnikai
K.F.T.

1073 Budapest VII., Erzsébet krt. 48.
Telefon/Fax: 142-2115

BASF
video- és audiokazetták
NAGYKERESKEDÉSÉVEL
is foglalkozik!

...nem csak számítástechnika

INFORMÁCIÓKÉRÉS: 12 ▲

SZÁMÍTÁSTECHNIKA KULCSRAKÉSZEN!

A LEGKISEBB NOTEBOOK-TÓL
A LEGGYORSABB 486-OSIG

- XT, AT, 386, 386SX, 486, Laptop minden kiépítésben.
- EPSON, STAR, NEC nyomtatók teljes választéka.
- MODEMEK és egyéb tartozékok széles választéka.
- Magánszemélyeknek KÉSZPÉNZFIZETÉS ESETÉN KEDVEZMÉNY!
- ASHTON-TATE, BORLAND, MICROSOFT, NANTUCKET, LOTUS szoftverek.
- SHAREWARE programok (1200-féle) 360,– Ft + ÁFA áron.
- MODEMES táv-adatátviteli és BBS rendszerek szállítása.
- VÍRUSÖLŐ program (120-féle vírust öl!)
- NOVELL HÁLÓZATI SZOFTVEREK, hálózatkiépítés.

Ajánlatunk:
NOTEBOOK SZÁMÍTÓGÉPEK MÁR 69 900,– Ft-tól

AT számítógép: 1 MB RAM, 40 MB HDD,
1,2 MB FDD, Mono 14" (PHILIPS)
1 S, 1 P, 101 gombos bill. 63 100,– Ft + ÁFA
(Készpénzért 59 900,– Ft + ÁFA.)

Amikor ezt a hirdetést Ön olvassa, áraink már úgyis alacsonyabbak! Ezért kérjük, telefonáljon vagy írjon, és mi örömmel adunk felvilágosítást, küldünk részletes árjegyzéket!

QWERTY

High Tech. Kft.
1117 Budapest XI., Orlay u. 4.
Telefon: 166-3098, 185-2687, Fax: 185-2687
BBS: 118-7950 BUDAPEST BBS

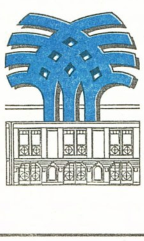
INFORMÁCIÓKÉRÉS: 24 ▲

FOXPRO — kontra nélkül

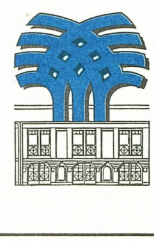
Ami miatt jobb, ha FOXPRO 2.0-t használunk:

— Egérvezérlés, kifinomult ablaktechnika
— Képernyő-, menütervezés, tablók készítése
— Automatikus programkód-generálás
— Összetett indexforma
— Projektkészítés
— Rushmore-technológia
— Trace, debug funkció
— .EXE állományok készítése
— Automatikus programdokumentálás

Rendszeres szakmai bemutatók!



Upgrade-lehetőség rendkívül kedvező áron!



FoxPro 2.0 (Egyfelhasználós)	31 500 Ft + ÁFA
FoxPro 2.0 (Többfelhasználós)	50 900 Ft + ÁFA

INFORMÁCIÓKÉRÉS: 14 ▲

Hozzáférés — két akadályon áthágva IV.

A jogosultságok technikája

Ebben a részben eljutunk a konkrét problémához: ki és mire jogosult a rendszerrel való kommunikációban? A kérdés és a válasz összetettségéről és ennek technikai vetületeiről lesz most szó. A PC-ken működő PC-DOS vagy MS-DOS a fájlokat hierarchikus felépítésű könyvtárrendszerben tárolja. Ebben a környezetben a következő fájl-, illetve könyvtárkezelő műveletek azok, amelyeket egy fájlszintű hozzáférés-védelemnek éberen kell figyelnie:
Fájlműveletek (pontosabban fájlokat érintő műveletek): létrehozás, törlés, átnevezés, megnyitás, olvasás, írás, a fájl attribútumának megváltoztatása, futtatás.

A futtatás fogalomnál nem az összes, ebben a környezetben elindítható fájlra gondoltunk, hanem csak az EXE, illetve COM kiterjesztésűekre. Ennek az az oka, hogy a BAT kiterjesztésű fájlok indításakor a DOS a fájlnyitás, -olvasás, továbbá programfuttatás vagy parancsvégrehajtás műveleteket végzi, vagyis gyakorlatilag semmi olyan új dolgot, amelyet ne kívánnánk egyébként is figyelni.

Könyvtárműveletek

Természetesen nem mindenki vehet abból a süteményből, amelyet megkíván, vagyis nem hajtathat végre minden felhasználó minden műveletet az adott géppel.

Egy jó hozzáférés-védelmen belül a jogosultságoknak személyhez köthetőknek kell lenniük. A személyhez való kötődést a számítástechnikában általában kulcsszóval oldják meg. PC-s környezetben a felhasználó—kulcs- szó—alkönyvtár—fájl viszonyt úgy képzelhetjük el, mint egy táblázatot, melynek első sorát a felhasználók képezik a hozzájuk tartozó kulcsszóval, első oszlopában pedig az alkönyvtárak és a bennük szereplő fájlok vannak. A táblázat bármely eleme az aktuális jogosultságot adja meg.

Mielőtt rátérnénk a „jogosultság" bővebb tárgyalására, adjunk nevet annak a személynek, akinek a hatáskörébe tartozik a felhasználó—kulcsszó—alkönyvtár—fájl viszonyrendszer meghatározása: nevezzük egyszerűen Főnöknek.

A jogosultságokról

A jogosultság különböző műveletek végzésére kapott (illetve adott) engedélyek összessége. A fentebb felsorolt fájlokat, illetve alkönyvtárakat érintő műveletek megtiltásakor/engedélyezésekor, valamint a figyelés megszervezésekor többek között a következő dolgokra kell gondolnunk:

— ha az EXE vagy COM kiterjesztésű fájlt a felhasználó olvashatja, akkor semmi nem akadályozhatja meg abban, hogy futtassa is;

— felesleges meghagyni a fájlnyitás jogát, ha a felhasználó sem nem olvashatja, sem nem írhatja át az adott fájlt;

— nem célszerű az írást engedélyezni olyan fájl esetében, amelyiknél az olvasást megtiltottuk;

— nem engedélyezhető egy fájl attribútumának megváltoztatása alkönyvtár jelzésűvé, illetve egy alkönyvtárénak standard fájl jelzésűvé;

— fájl létrehozásának jogát csak alkönyvtárra vonatkozóan engedélyezhetjük, illetve tilthatjuk meg (egy még nem létező fájlhoz nem tartozhat jogosultság);

— ha az alkönyvtár-struktúrában egy „levelet" elérhetővé akarunk tenni, akkor az odáig érő „ágakat" sem vághatjuk ki;

— felesleges izgatni a felhasználót azzal, hogy az alkönyvtár tartalmának listázásakor láthat ugyan egy bizonyos fájlnevet, de a fájl nem hozzáférhető a számára.

Ezek figyelembevételével a következő jogosultsági lehetőségeket dolgoztuk ki a fájlokra vonatkozóan: láthatóság (lásd az előző felsorolás utolsó pontját); olvashatóság; írhatóság; attribútum megváltoztatása; futtathatóság; törölhetőség; átnevezés.

(A legutolsó pont megoldása egy kicsit problémás, ugyanis a fájlokat a nevükkel azonosítjuk, és ehhez kapcsolódnak a különböző jogosultságok. Ezért a név megváltoztatásával együtt komoly módosítások szükségesek a későbbiekben ismertetendő táblázatokban.)

Az alkönyvtárakra vonatkozó jogosultsági lehetőségek: láthatóság (ez azt jelenti, hogy az alkönyvtár nevét a DIR parancs megmutatja, és minden további nélkül lehetőség van a „belelépésre"); fájl, illetve „gyermek" alkönyvtár létrehozása; törölhetőség.

Léteznek továbbá nem fájl- vagy alkönyvtárszintű jogosultságok is, amelyekről feltétlenül szólnunk kell e helyen is — annak ellenére, hogy az 1. lépcső kezeli le ezeket. Ilyen a hajlékonylemez olvasása, írása, formázása, valamint a printer kezelése.

Ezeknek a figyeléséről az 1. lépcső gondoskodik, de arról, hogy az adott műveletet engedélyezheti-e vagy sem, a 2. lépcső értesíti.

Technikai megoldások

Miután a Főnök minden egyes felhasználóhoz hozzárendelt egy kulcsszót, valamint a fájloknak, illetve az alkönyvtáraknak egy, az adott felhasználó által elérhető halmazát és az ezekhez tartozó jogosultságokat, már komoly szerepet kap a szűrő 2. lépcsője. Az eddigiekben láthattuk, hogy milyen feladatokat kell ellátnia ennek. Itt az ideje, hogy megnézzük, megoldhatóak-e ezek a feladatok, és ha igen, akkor hogyan.

Amikor a felhasználó valamilyen műveletet akar végezni egy fájllal vagy alkönyvtárral, de ehhez nem rendelkezik a megfelelő jogosultsággal, akkor

meg kell akadályozni ebben. Ha a Főnök történetesen ott áll az illető mögött, és látja, amit csinál, akkor lecsaphat rá és megdorgálhatja. Ellenkező esetben a 2. lépcsőnek kell közbelépnie és visszaterelnie a karámba az eltévedt báránykát. De honnan tudhatja a szűrő, hogy mi az, amit szabad a felhasználónak, és mi az, amit nem? A felhasználók kulcsszavait és a hozzájuk tartozó jogosultságokat táblázatokban kell tárolnunk. Ezek célszerű felépítése a következő.

Kulcsszavak

Ofszet és mérettartalom

+00—1 — a kulcsszavak száma (1 bájton maximum 255; nem ez a felhasználók számának a korlátja)

+01— a táblázat sorainak felépítése (egyenként 18 bájton):

+00—16 — kulcsszó

+17—1 — a felhasználó nem fájlszintű jogosultságainak (floppyolvasás, -írás, -formattálás, printerkezelés) kódja

+18—2 — a kulcsszóhoz tartozó jogosultsági táblázat kezdő clustere.

A táblázat sorainak száma, vagyis az, hogy hány felhasználói kulcsszó lehet a rendszeren belül, a lemez clusterenkénti szektorszámától függ (ugyanis 1 clusternyi lemezterületet szántunk erre a célra). Egy szektor 512 bájtból áll, egy cluster pedig 2, 4, 6 vagy 8 szektort foglalhat magában. A tényleges értéket a bootszektor tartalmazza. Ettől az értéktől függően a következőképpen alakul a tárolható kulcsszavak maximális száma:

Szektor/cluster — kulcsszavak maximális száma

2—53
4—107
6—161
8—215

Bármelyik eset áll is fenn, nem valószínű, hogy ez ténylegesen korlátot jelentene. (Már 20 felhasználó is irreálisan sok egyetlen PC-re, nem még 56.)

Jogosultsági táblázat

Az előző táblázat minden sorához tartozik egy ilyen táblázat. Ez már több clusterből is állhat. Felépítése a következő:

Ofszet és mérettartalom

+00—2 — a táblázat következő clusterének a sorszáma (a vége jel a 0 érték)

+02—2 — a táblázat sorainak száma a clusteren belül (tele cluster esetén a maximális érték)

+04 — a táblázat sorainak felépítése:

+00—2 — a bejegyzés clustere

+02—1 — a bejegyzés sorszáma a clusteren belül

+03—11 — a fájl neve

+14—1 — a fájl eredeti attribútuma

+15—2 — a fájl kezdő clusterének sorszáma

+17—3 — a jogosultság kódja

Láthatatlanná tenni

Felmerülhet az olvasó fejében, hogy miért tároljuk a fájlok nevét, attribútumát és első clusteruk sorszámát is, amikor bőven elég lenne a bejegyzést tartalmazó cluster sorszámának, a bejegyzésclusteren belüli sorszámának, valamint az aktuális jogosultságnak a feljegyzése. A válasz a következő: két legyet ütünk egy csapásra, ha a kezdeti jogosultságfeltöltést (erről még lesz szó a későbbiekben) összekötjük a láthatóság problémájának megoldásával.

A láthatóság problémájának megoldása komoly feladatnak ígérkezik, ha figyelembe vesszük, hogy milyen erős és bölcs ellenfeleket kell becsapnunk. Egy PC Toolsnak vagy egy Norton Commandernek semmiféle akadályt nem jelent egy rejtett (hidden) jelző a fájl attribútumában. Ezzel szemben mi mégis azt szeretnénk, hogy az adott felhasználónak bizonyos fájlok létezéséről ne legyen tudomása.

Ha a titkolandó fájl fájlbejegyzésében a fájlnév első pozíciójára egy '.' karaktert írunk (a karakter ASCII kódja hexadecimálisan 2E) és a további pozíciókat nulla értékekkel töltjük fel, akkor semmiféle fájlkezelő nem jelzi ki a bejegyzés tartalmát. De sajnos a kíváncsi felhasználó (vagy a még kíváncsibb tolvaj) átírhatja a fájlnév törölt pozícióit (mondjuk PC Toolsszal), és már közelébe is jutott a féltett adatoknak. Ezért szépen kinullázzuk a bejegyzésben a fájl (vagy alkönyvtár) első clusterét jelző sorszám helyét, és a biztonság kedvéért a fájl attribútumát tartalmazó bájtot is. Ezzel teljesen elzártunk minden utat az illetéktelen szemek elől.

Abban az esetben, ha a felhasználónak joga van látni a kérdéses fájlt (minden bejegyzésbeli adatával együtt), egyszerűen visszaírjuk azokat a megfelelő helyekre. Ezt a visszaírást végezzük a jogosultságkódok beírásával egy időben. (A későbbiekben még szólunk arról, hogy hová is írjuk be a jogosultságkódokat).

Ezek alapján nyilvánvaló, miért jelent gondot egy fájl átnevezése: ezt a változást minden egyes, kulcsszóhoz tartozó jogosultsági táblázatba át kell vezetnünk.

A láthatóság problémájának ez a megoldása veszélyt is rejt magában: ha egy adott clusterhez nem tartozik bejegyzés (és esetünkben látszólag ez a helyzet a titkos fájlok tartalmánál), vagyis a cluster „árva", akkor egy óvatlanul kiadott CHKDSK parancs vagy a SpeedDisk nevű Norton-utility használata tönkreteheti a merevlemezen lévő adatokat.

Ezt úgy oldhatjuk meg, ha ilyen jellegű műveletek végzésének jogát csak a Főnök kezében hagyjuk meg. Ő ugyanis az összes fájlt, illetve alkönyvtárat illetően minden jogosultsággal rendelkezik. Ez azt jelenti, hogy amíg ő dolgozik a gépen, addig az összes fájl látható állapotban van, vagyis nincs olyan „árva" cluster, ami a „láthatatlanság" következménye lenne.

A jogosultság kódja

A fájlnevek, illetve jogosultságok beírása után már „csak" az a feladat van hátra, hogy a megfelelő pillanatban a megfelelő helyről kiolvassuk az adott jogosultság értékét, és ennek megfelelően engedélyezzük, illetve tiltsuk az aktuális művelet végrehajtását.

Volt már szó arról, hogy a fájlokhoz és alkönyvtárakhoz tartozó jogosultsági kódokat első lépésben hogyan tároljuk, de azt eddig rejtélyesen elhallgattuk, hogy hová is tesszük majd ezt a kódot, ahonnan a lehető leggyorsabban beolvasható. Nos, ez a hely a fájlbejegyzés maga. Az alapfogalmaknál láthattuk a fájlbejegyzés szerkezetét, és ebben bizony akadt — az általunk ismert DOS-verziók által még kihasználatlan — 10 bájt. Ide bőven elfér a jogosultsági kód 3 bájtja.

Persze felmerül a kérdés, hogy mikor tölthetjük fel ezt az ominózus 10 (pontosabban 3) bájtot a megfelelő értékkel, ugyanis ennek különböző felhasználók esetén különböző tartalommal kell bírnia. Erre a legalkalmasabb időpont a kulcsszó lekérdezése (loginolás), valamint annak helyes eredménnyel záródó kiértékelése után kínálkozik. Ekkor a bejelentkezéssel egy időben az általunk kódolt formában tárolt táblázatból feltöltjük az „üres" helyeket (beírjuk a jogosultságok kódjait és a látható fájlok eredeti adatait), és a védelmi rendszernek ez a lépcsője már működhet is.

Krokovay Károly—Radványi Tibor

Alapvetően **ÚJ!** koncepció

LAN-GUARD

Integrált hálózati biztonsági rendszer

Lokális hálózatok

VÍRUS- és ADATVÉDELME

- FILE-SERVER-ek, terminálok hozzáférés- és bootvírus-védelme.
- File-vírusok elleni védelem.
- Integrált munkafolyamat-vezérlés.
- Egyedi számítógépek védelme.

Ha fontosak az adatai, segít a

DATA DOCTOR Kft.
1149 Budapest, Buzogány utca 4.
Telefon/Fax: 18-37-299

INFORMÁCIÓKÉRÉS: 23 ▼

SZÁMÍTÁSTECHNIKA KULCSRAKÉSZEN!

- AT 286/386/486 igény szerinti kiépítésben
- EPSON nyomtatók és kiegészítők teljes választéka
- Hardveralkatrészek nagy választékban
- Hálózattervezés, -építés
- Átalánydíjas és eseti szerviz

Készpénzes ajánlatunk:
Alapkonfiguráció: alaplap, RAM, 1,2 Mbájt FDD, 40 Mbájt HDD, IDE, multi I/O, MGP, 14" mono monitor, Baby-ház, 101 gombos billentyűzet.

AT 286-12/16 + 1 Mbájt	58 800,–
AT 286-16/21 + 1 Mbájt	60 800,–
AT 286-20/25 + 1 Mbájt	74 800,–
AT 386-25/32 + 2 Mbájt	103 100,–
AT 386-33/54 + 64 kB Cache + 2 Mbájt	122 000,–
AT 386-40/65 + 64 kB Cache + 2 Mbájt	139 000,–
ISA 486-25/114 + 64 kB Cache + 4 Mbájt	190 400,–
ISA 486-33/150 + 256 kB Cache + 8 Mbájt	232 400,–

Kérje részletes árlistánkat!

Áraink 12 hónap cseregaranciával, ÁFA nélkül értendők.

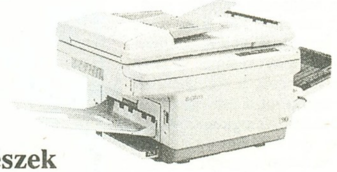
UNITRADE
Szervezési, kereskedelmi és Számítástechnikai
K.F.T.
1073 Budapest VII., Erzsébet krt. 48.
Telefon/Fax: 142-2115

...nem csak számítástechnika

INFORMÁCIÓKÉRÉS: 27 ▼

REX TRADE

Kereskedelmi és Szolgáltató Kft.



Másológépek

Kellékek

Festékek



Alkatrészek

A/3, A/4, A/5 papírok

Pauszmásológépek

Telefaxok

Cím: REX TRADE Kereskedelmi és Szolgáltató Kft.
1139 Budapest Fáy u. 6.
Telefon: 1203-280/149-es, 156-os mellék, 1202-805

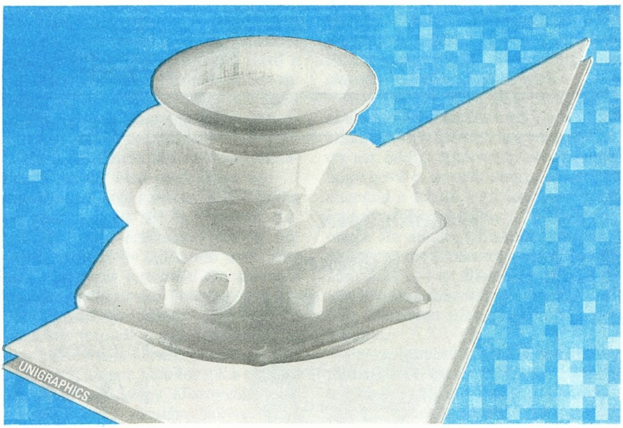
INFORMÁCIÓKÉRÉS: 17 ▼

Lézerlitográfia

Testmodellező szoftverek

A számítógépes tervezés és gyártás egymásra épülését jól illusztrálja a lézeres litográfia elnevezésű eljárás, amely bonyolult műanyag alkatrészek prototípusának előállítását teszi lehetővé.

Tudjuk, hogy az akár több száz műanyag alkatrészt is tartalmazó termékek piaci életgörbéje olykor csak egy-két év. Például egy fényképezőgép vagy videokamera befuttatására alig néhány hónap áll rendelkezésre, ugyanakkor a tervezési-gyártási költségek milliókat emésztenek fel. Így a termékek tervezésénél nem szabad tévedni. Ezt igyekszik elősegíteni a lézeres sztereolitográfia, amely a prototípus- és kissorozatgyártás forradalmian új módja.



Prototípustól a gyártásig

A lézerlitográfiás berendezés tartalmaz egy kb. 50 x 50 cm felületű, folyékony műgyantával feltöltött kádat, amelyben a sűrű folyadék felszínét tűhegyes lézersugár pásztázza. A 0,1 mm vastagságú, ultraibolya tartományban működő lézersugár egy felületet rajzol körül, és azt ki is „satírozza". A végigsatírozott felület alatt a műgyanta helyileg megkeményedik, mintegy két tized mm mélységben. Így készül el a kialakítandó test egy keskeny szelete. A már megkeményedett részt a folyadékban levő befogó szerkezet a szelet szélességével lesüllyeszti. A megkeményedett felület helyére újra folyékony műgyanta folyik. Ha ez megnyugodott, a lézersugár megkeményít egy újabb szeletet, amely hozzáköt a már elkészülthöz. Így szeletenként alakul ki a teljes test, ahol egy-egy felület lézeres megvilágítása néhány percet, míg a teljes test kialakítása néhány órát vesz igénybe.

A folyékony műgyanta összetétele alapján lesz kemény vagy rugalmas az elkészült műanyag alkatrész. Bár a speciális műgyanta nagyon drága — kilogrammonként néhány száz dollár — azonban egy-egy alkatrészhez viszonylag keveset használnak fel. A kész alkatrészeket a tervező akár már ellenőrizheti is. A termékfunkció analízisekor pedig működési, szilárdsági, illesztési és szerelhetőségi hibák is kiderülhetnek.

Modellezés — tervezés

Háromdimenziós testmodellező szoftverrel generáljuk a lézerlitográfiás berendezés vezérlőprogramját. A térben elhelyezett, összetett alkatrészek (testek) különböző nagyságú és pozíciójú elemi testekből épülnek fel (tégla, gúla, kúp, gömb, tórusz). Ezekből halmazműveletekkel (testek összeadása, kivonása, közös rész képzése, testek síkokkal való metszete) állíthatók elő a bonyolult üreges testek. Egyes testmodellező szoftverek megengedik az összeépített testek egyes felületei között kialakult élek lekerekítését is. Ezek a szoftverek elsősorban Unix-alapú gépeken futnak, de egyszerűbb feladatok PC-s környezetben is elvégezhetők. Mivel a testmodellező szoftverek más-más belső adatbázist használnak, így a lézerlitográfiás berendezés számára a teseket szabványos formátumban kell leírni.

Elterjedt módszer a felületek behálózása. A hálót tovább finomítják, és a kapott elemi felületdarabokat síkbeli háromszögekre bontják. A háromszögekre bontás szokásos rutineljárás. Mivel egyes nagyfelbontású grafikus kártyák is háromszögekből jelenítik meg — színesen, valósághűen — a testeket, így ugyanaz az adatbázis használható a tervezett test képernyős ellenőrzésére is. A kapott háromszögekből a lézerlitográfiás berendezés rekonstruálja a testet. Az előírt technológiai vastagságnak megfelelően felszeleteli és a kapott szeleteket besatírozza a kapott lézersugár szélességének megfelelően.

Az alkalmazott technológia legnagyobb előnye, hogy segítségével tetszőleges bonyolultságú és tetszőlegesen sok üreget tartalmazó prototípus-alkatrészek gyárthatók. Ha a prototípus megfelel, akkor ugyanezen adatbázisból gyakorlatilag emberi beavatkozás nélkül készülnek el a szerszámgyártó NC-berendezések (pl. marógép) vezérlőadatai. Úgy érezzük, ez a technológia egy újabb, jelentős lépés a számítógéppel vezérelt, integrált tervező-gyártó rendszerek kifejlesztése irányába.

Balázs János

Új szerszám a grafikusoknak

Dekoráljunk CAD-élyesebben!

A CAD sokszínű világából most egy hazai viszonylatban új dekorációs technológiát, és hozzá a gyakorlatban jól használható, itthon is kapható rendszert ismertetünk.

A hazai piacon nemrég megjelent számítógép-vezérlésű grafikai, fóliavágó és gravírozó rendszerekkel először a DekoCAD Kft. termékbemutatóján találkoztunk, de forgalmazza azokat több más cég, például a Mikropo is. Ezeket a komplett rendszereket jól használhatják a számítástechnikához keveset értő reklámszakemberek, dekoratőrök, címfestők vagy „szitások". Az ollóval, ecsettel, ragasztóval végzett manuális tevékenységsorozatot váltja ki a számítógépes dekoratőr.

Szkenner vagy digitalizáló tábla?

A dekorációs rendszer lelke egy modulokból álló és magyarul „beszélő" program, amellyel különféle feliratokat és grafikákat készíthetünk, illetve feldolgozhatjuk a már meglevő rajzokat, emblémákat. A rendszerhez tetszőlegesen kapcsolható szkennerrel vagy digitalizáló táblával beolvashatjuk a már meglevő ábrákat, grafikákat, fényképeket. A képeredeti minőségének megfelelően választjuk ki az eszközt: jó minőségű, de bonyolult grafikai elemeket célszerűbb beszkennelni, míg a gyengébb minőségűekhez jobb a digitalizáló táblát használni. A választást befolyásolhatja például a kép „sarkos" vagy „ívelt" kontúrja is. Mindkét esetben megoldott azonban az egyenesek és a görbék törésmentes egymáshoz illesztése.

Ha a beolvasás és a szükséges feldolgozás után az előállított ábrákkal és feliratokkal elégedettek vagyunk, azokat rögtön ki is vághatjuk vágógéppel valamilyen anyagból, leginkább öntapadós fóliából. Vastagabb fólia, papír, karton, bőr és textil kivágásához vágóasztal is kell. Ha a vágóasztalt gravírozó, illetve kontúrmaró fejjel egészítjük ki, sokféle más anyag (pvc, plexi, fa, réz) is megmunkálható.

Sign Tronic, a vágószoftver

A program a betű típusának és méretének megadása után a szöveget kiírja a képernyőre. Ezt a feliratot a továbbiakban sokféleképpen variálhatjuk: dönthetjük, nyújthatjuk, zsugoríthatjuk, csavarhatjuk, mégpedig nemcsak az egész szöveget egyszerre, hanem annak kisebb részeit külön is. A feliratokat és részeiket elforgathatjuk, színezhetjük, csíkozhatjuk stb. Speciális utasítással a szöveget áthelyezhetjük egy másik ábrába, ahol további szövegmanipulációkat végezhetünk. Lehetőségünk van a betűk és ábrák egymásba lógatására, a betűközök meghatározására.

A program a szöveget tetszés szerinti vonalhoz igazítja, kör alakban kiírja, gömb vagy henger felületére történő illesztést imitál stb. Kaphatnak a betűk, ábrák kontúrt, árnyékolást is. Egyszerű árnyékolás esetén a nem látható éleket a plotter nem vágja ki. Dobozárnyékolással például könnyedén készíthetünk térhatású feliratozást. A saját feliratok mellé pedig más programokból is átvehetünk kész feliratokat, emblémákat, grafikákat.

A szerkesztési műveletek után következhet a kivágás. Az öntapadó fóliára elkészített feliratot egyben és egyszerre felragaszthatjuk a végleges felületre. Egyszínű feliratoknál a dekorációs anyag előkészítésének egész folyamata rendkívül leegyszerűsödik. Többszínű feliratok, emblémák esetében az eljárás valamivel bonyolultabb, mivel az egyes színekből külön-külön kell ki vágni az azonos színű grafikai elemeket, de azokat a program által megjelölt illesztési pontok mentén gond nélkül összeállíthatjuk. Ezt a technológiát azok a dekoratőrök értékelik majd leginkább sokra, akiknek gyakran kell bonyolult ábrájú és többszínű reklámfeliratot készíteniük.

A hálózatban is működő program — amely a Sign Tronic cég fejlesztése — DOS alatt fut, magyarul kommunikál, és a hazai szokásoktól eltérően magyar nyelvű felhasználói kézikönyv is segíti megértését és használatát.

Sziebig Andrea





MACRODA – A MODERN SZÁMÍTÁSTECHNIKA!

1992-ben is kínálunk:
„THE MACRO" számítógépeket
1+2 év garanciával,
CAD rendszereket,
3 M mágneses adathordozókat,
mágneskártyás adatvédelmi rendszereket,
számítástechnikai kiegészítőket,
STAR és CITIZEN nyomtatókat,
CANON irodatechnikát.

MACRODA KFT.
MINTABOLT: 1123 Bp., Alkotás u. 21.
Tel./Fax: 156-4802
IRODA: 1016 Szirtes u. 28./A
Tel.: 186-5782, 186-5686, 185-7866
Fax: 186-5686, Telex: 22-5375

MACRODA

Kérje részletes árlistánkat!

INFORMÁCIÓKÉRÉS: 06 ▲

NTT - 2000 RANK XEROX

TRADE and SERVICE Ltd.

HIVATALOS DEALER
és MÁRKASZERVÍZ

1103 Budapest, Gyömrői út 86. Telefon: 147-2734, 147-2735 Telefax: 147-2301

RANK XEROX





MÁSOLÓGÉPEK,	KELLÉKANYAGOK
MÁSOLÓPAPÍROK,	FÓLIÁK
FAXPAPÍROK,	LÉZERNYOMTATÓ
TELEFAXOK,	ÍRÓGÉP

RANK XEROX MÁRKASZERVIZ

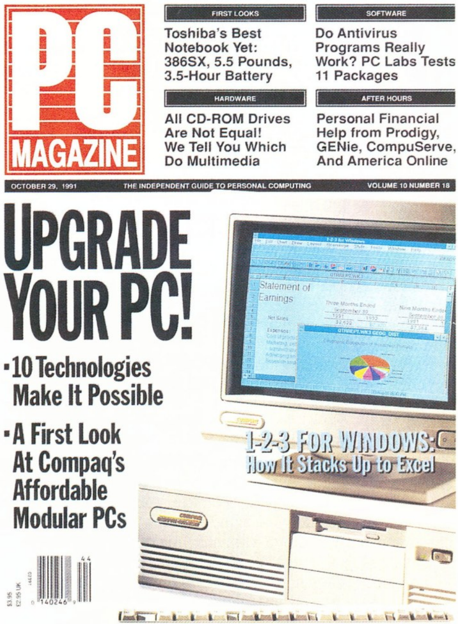
— Helyszíni üzembehelyezés
— Karbantartás
— Garanciális szolgáltatások
— Garancián túli javítás, karbantartás
— Háromféle szerződéstípus

NTT — 2000 Kft.

PARTNER AZ IGÉNYESSÉGBEN!

INFORMÁCIÓKÉRÉS: 07 ▲

Microsoft kontra IBM — viszály vagy álviszály



Bill Gates hirhedtté vált feljegyzésével kezdődött — amelyet a San Jose Mercury News szivárogtatott ki először. Ebben Bill Gates az IBM-et azért marasztalja el, mert a Microsoft részére átadott kód, enyhén szólva pocsék. Továbbá azt is tudtul adja, hogy a Microsoft az IBM nélkül valójában jobban érzi magát.

A legtöbb megfigyelő szerint ez a feljegyzés a két cég ellentéteinek kiéleződésére utalt, ami körülbelül akkor kezdődött, amikor kiadták a „bogár-rágta" (bug-ridden) DOS 4.0-át. Később a feszültséget a Windows 3.0 sikere tovább gerjesztette.

Mások szerint azonban a viszálykodás esetleg csak színjáték, ami része lehet egy bonyolult tervnek, valami nagy átverésnek vagy összeesküvésnek. Vizsgáljunk meg négy lehetséges forgatókönyvet.

1.) A „látszat-szakítási" elmélet

Az IBM és a Microsoft a viszálykodás látszatát kelti. Ha a két cég látszólag eltávolodik egymástól, beszivároghatnak a versenytársak közé, és új termékeikkel túljárhatnak az eszükön. A Microsoft az ACE csoport oszlopos tagja, így nem kellene sok ahhoz, hogy a csoportot félrevezesse (Compaq, DEC, MIPS, Silicon Graphics és 60 egyéb). Egyedül az operációs rendszerek szállításának visszatartása több milliós veszteséget okozna.

Eközben az IBM az Apple-lel bújik ágyba, ami megfelelő pozícióba juttatja ahhoz, hogy az Apple-t kijátszhassa. Bár az IBM soha sem tárná fel az Apple-lel fennálló kapcsolatának összes titkát, az is épp elég, ha átadja a Microsoftnak a jövőbeli termékek specifikációját, kiszolgáltatva ezzel, hogy az Apple éppen mivel foglalkozik.

2.) A „Bill 'Machiavelli' Gates" elmélet

Bill Gates grandiózus bábmesterként, saját hasznára, álviszályt keltett a két cég között. Először is kivette az IBM kezéből az MS-DOS feletti hatalmat, hogy a DOS-t ugyanúgy egyenesen a végfelhasználó részére árulhassa, mint ahogy azt gyilkos sikerrel a Windows esetében megtette. Azután a későbbi beavatkozások lehetőségének elkerülésére az IBM-et teljesen ki kell iktatnia a képből. A Microsoft nyugodtan ösztönözte az IBM–Apple partneri kapcsolatot, mivel tudta, hogy az IBM saját tenyésztésű „apparátcsikjai" soha sem lesznek képesek együtt dolgozni az Apple vezetőivel. Mindkét cég rengeteg pénzt költ majd el, de semmilyen eredménye nem lesz ennek a partneri kapcsolatnak. Az elmélet szerint Gates az IBM-et és az Apple-t alacsony beosztású tisztviselők színfalak mögötti manipulációjával kapcsolta össze.

3.) A „kísérlet az Intel megrémisztésére" elmélet

A Microsoft mindig szorosan kapcsolódott az Intelhez, és hogy a versenytársakkal szemben előnyhöz jusson, jó, ha ismeri a mikroprocesszorok következő generációjának belső felépítését. Azt is fontos megtudnia, hogy melyek lehetnek a legnépszerűbb mikroprocesszorok — ne kelljen nagy vargabetűvel például az RS6000 felé fordulnia, miután már irányt vett a MIPS-re. Az Intel azonban nem veszi komolyan a RISC-et. Bár az Intelnek megvannak az erőforrásai, hogy a CISC csip architektúrát a végtelenségig nyomja előre, a CISC-et állandóan versenyre fogja hívni egy-egy RISC csip. Ha az Intel végre lenyelné a keserű pirulát, és a RISC irányába mozdulna, nyerő helyzet alakulna ki. A legjobb módja annak, hogy az Intelt eszméletre térítse, ha a Microsoft szintén belép az ACE paktumba, és ráveszi az IBM-et és az Apple-t is, hogy a RISC-et válasszák jövőbeli architektúráikban. Az Intel ezt már nem nézhetné tétlenül.

4.) A „Gates-t ez nem érdekli" elmélet

Bill Gates egy nagyon érdekes grafikont mutatott be a tavalyi Comdexen: a PC-piac laposodó növekedési görbéjét. Hogy a Microsoft „forró cég" maradhasson, továbbra is terjeszkednie kellene, mégpedig egy olyan piacon, amely már nem növekszik tovább. Az operációs rendszerek piaca volt az, ami miatt a Microsoftnak szüksége volt az IBM-re. Mára a telepített klónok és IBM gépek részére az új verziókat az IBM nélkül is eladhatja. Gates azt is tudja, hogy a növekedés fenntartása érdekében, a cégnek ezentúl valamennyi piaci szegmens részére alkalmazásokat kell eladnia, és ez nyílt versengést jelent az IBM-mel. Versengeni viszont könnyebben lehet, ha a versenytárs nem ugyanabban a házban lakik, és nem befolyásolja a döntéseidet. E szerint az elmélet szerint Gates őszinte volt, amikor kijelentette, hogy cége az IBM nélkül jobban megvan.

(PC Magazine, 1991/17)

Programozási eszperantó

Azt pletykálják, hogy az IBM végre azt az OS/2-t fogja kiadni, amelyikre a világ már régóta vár: egy valódi multitaszkos rendszert, amely futtatja a 32-bites Presentation

Manager (PM) alkalmazásokat, a 16-bites PM és a karakteres OS/2 alkalmazásokat, a Windows alkalmazásokat, a DOS alkalmazásokat, és még a DOS eszközmeghajtókat is.

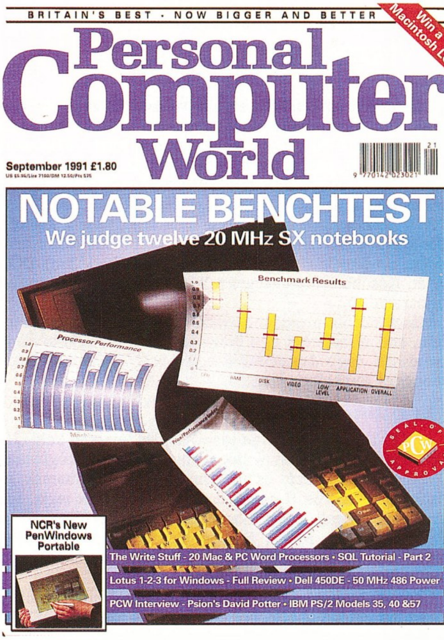
Az elkészült OS/2 2.0 Microsoft Software Development Kit verzió nagyon érdekes, de az IBM ígéreteitől még nagyon messze van. A fejlesztők ezért nehéz döntés előtt állnak: hogyan készítsenek olyan szoftvert, amelyet nem kell minden új operációs rendszerre és felhasználói interfészre újraírni?

Nem is csak a Windows és a Presentation Manager a cél. Olyan egyszerű alkalmazásokról van szó, amelyek bármelyik grafikus felhasználói felületen futnak (GUI = graphic user interface), és közben olyan szépek és jók, mintha gondosan éppen arra a környezetre írták volna őket. És nem is csak DOS és OS/2 operációs rendszerű gépeken, hanem Apple Macintoshon, Amigán, Atari ST-n vagy valamelyik tetszés szerint választott Unix gépen. Ez talán csak álmodozásnak tűnik, de ha szoftverfejlesztő cégünk kicsi, nem engedheti meg magának azt a luxust, hogy a kódot minden környezetre újra átdolgozza.

Egészen mostanáig az emberek a cégüket egy lapra, egyetlen gépi környezetre tették fel. Ha vesztettek, újra kezdték egy másik környezetben, és közben abban reménykedtek, hogy ott képesek lesznek legalább addig kitartani, amíg adósságaikból kikerülnek.

Aki komolyan gondolja, hogy több környezet részére írjon, annak a kódszétválasztás (code segregation) gyakorlatához kell folyamodnia. Sajnos, ilyenkor több fegyelemre van szükség. Ahhoz, hogy a Windows-hoz írt programok forrásszinten hordozhatóvá váljanak, makrókat kell létrehozni az egyszerűbb konverziók elvégzésére, a bonyolultabb esetekhez szubrutinokat, a legrosszabb esetekben pedig a rémes #ifdef blokkokat kell használni.

(Byte, 1991/augusztus)



Ventura 4.0 — színek az Ablakban

A Ventura Publisher a 2.0 verzió megjelenése óta jelentős változásokon ment keresztül, főleg a Windows és a Macintosh környezethez történő alkalmazkodásnak köszönhetően. A Ventura 3.0 GEM- és Windows-változata közötti minimális különbségek inkább még csak az eltérő grafikus felületből adódtak, a GEM korlátai miatt azonban várható volt, hogy a Windows-Ventura előbb-utóbb kiugrik, lehagyja társát.

Mindez a Ventura 4.0 Windows-változatában már valósággá is vált. A korábbi Ventura programokba például nem volt beépítve helyesírás-ellenőrzés, ezt a műveletet ezért a Ventura keze alá dolgozó szövegszerkesztőkben kellett elvégezni, így a belső formátumú szövegfájlok (.CAP) az ellenőrzésből kimaradtak. A Houghton Mifflin által készített és egyebek között az új Venturához is felhasznált AHED (American Heritage Electronic Dictionary) szótár 5 változata közül a legkisebb mérete 600 kilobájt (12 ezer címszóval), a legnagyobb pedig 8,1 megabájt (83 ezer címszóval). Az utóbbiban az elválasztási pontok jelölésén túl számos más információ is van (azonos alakú szavak, szóeredet, szógyökerek, használati megjegyzések), az elválasztási algoritmus pedig nagyon korrekt.

Egy másik Mifflin-program, az International CorrectSpell, 14 nyelven nyújt helyesírás-korrekciót, s különbséget tud tenni a brazil és az ibériai portugál vagy a német és a svájci német között is. A szótár felismeri a helytelen szóösszetételeket, tetszőlegesen bővíthető és a rendhagyó írásmódra is „megtanítható".

Az előző Ventura-változatokban fejezetenként összesen 2 nyelv volt megadható, s azok is csak az elválasztást oldották meg. A 4.0 tetszőlegesen bővülhet több nyelv szótárával, és a helyesírást is ellenőrizni tudja. (Érdekes módon a WordPerfect 5.0 változatával készült szövegeket elfogadja, de az 5.1 már nem tetszik neki.) A 3.0 változatban csak a kísérőszínes részek (spot colors) elkülönítése volt lehetséges, a 4.0-val már a többszínnyomáshoz szükséges színrebontás is elkészíthető (cián, bíbor, sárga, fekete).

Az alaptermékhez számos kiegészítő szoftver készül, amelyekkel együtt komplett Ventura Publisher dokumentációs környezet jön létre. A Ventura Database Publisher például önálló csomag, GEM és Windows alá, amely alkalmassá teszi a Venturát a .DBF állományok, továbbá a vessző- vagy szóközelhatárolású, illetve ASCII adatállományok fogadására, s azokból táblázat generálására.

A Windows alatt működő Ventura Publisher 4.0 színes képfeldolgozásához 4 további kiegészítő program tartozik: a Scan, a PhotoTouch, a Separator és a ColorPro Windows. Mind a négynek jót tesz, ha van a gépben koprocesszor, a ColorPro pedig 6 MB memóriát igényel.

— A Ventura Scan integrálódik az alapprogramba, és lehetővé teszi, hogy egy képet közvetlenül beszkenneljünk egy adott kiadvány tetszőleges oldalán lévő tetszőleges keretbe, majd ott még a szkennelési fázisban egészen ügyes színkorrekciókat és tónusváltoztatásokat hajtsunk végre.

— Képfeldolgozáskor a PhotoTouch 24-bites színekkel dolgozik, és használni tudja az összes elterjedt színskálát. A munka gyorsítására a képmanipuláció a képernyőn kis fel-

bontásban megjelenített képen történik, és utána kerül sor a teljes kép felfrissítésére. A precíz retusáláshoz körvonalkészítési, maszkolási lehetőség van.

— A Separator programmal készíthető el az oldalak színrebontása, úgy, hogy az illusztrációk és a szövegek már a nyomtatásnak megfelelően végleges helyükön vannak. Ebben a fázisban is lehetőség van a tónusértékek megváltoztatására, színkorrekcióra, a képek kontrasztosabbá vagy lágyabbá tételére.

— A ColorPro program egészen bonyolult színmanipulációkat, denzitométer-használatot, tónusérték-átszámítást stb. tesz lehetővé.

Összességében a Ventura Publisher 4.0, a hozzá tartozó egyéb programokkal együtt hatékony alkotási környezetet teremt és hasznos fejlesztéseket tartalmaz.

(Personal Computer World, 1991/12)

Mielőtt feltöltenénk címtárunkat...

Az adatbázisok leggyakoribb használati területe minden bizonnyal a postázási címlisták összeállítása, nevek és címek tárolása. Az alkalmas forma megtalálása egyszerű, ha az adatokra ugyanolyan formában van szükségünk, ahogy azokat a gépbe bevittük. De a helyzet máris bonyolultabb, ha egyéni megszólítással akarunk leveleket kiküldeni, ha csak borítékokat szeretnénk címezni vagy címkéket készíteni stb. Ezért sok későbbi gondtól megkímélhetjük magunkat, ha előre átgondoljuk adatbázisunk felépítését, mert könnyebb módosítani azon, ami már megvan, mintsem egyenként pótolni valamit, ami nem került rá az adatlapokra.

Először is azt kell persze eldöntenünk, hogy mit akarunk belevenni. Nagy a kísértés, hogy a kelleténél többet, „hátha szükség lesz rá" alapon. Csak azt felejtik el sokan, hogy utána ezeket az adatokat mindig fel is kell frissíteni, különben az egész adatbázis nem sokat ér, az pedig elég munkaigényes, ezért rendszerint nem jut rá energiájuk. Veszélyes a másik véglet is, hogy elegendőnek véljük a nevet és a címet, s csak jóval később jutunk el az akkor már rendszerint költséges felismeréshez, hogy más információkra is szükségünk lenne. A fenti két csapda elkerülésének legegyszerűbb módja, hogy egy egészen kis mintával az egész folyamatot lefuttatjuk, az adatbeviteltől a lehetséges felhasználási igényekig.

A magán jellegű címlisták szerkezete általában egyszerű: egy személy, egy cím, egy adatlap (record). Ha azonban vállalati, intézményi kapcsolatrendszerünket akarjuk gépre vinni, azonnal dilemmába ütközünk. Ugyanannál a cégnél több emberrel kapcsolatban állva legyen-e mindegyiknek külön adatlapja, és ismételgessük-e a címeket, vagy pedig a cég adatlapjához és címéhez rendeljünk több nevet, rovatot, esetleg telephely szerinti alrovatokkal és telefonszámokkal? Az első variáció felduzzaszthatja az adatbázist és gátolhatja az adatok frissen tartását, a másik pedig mindaddig jól működik, amíg el nem fogynak a rovatok, vagy pedig szükség lenne egyenként is a „közösben" tárolt információra. Esetleg kétszintű megoldáshoz lehet folyamodni: minden cégnek legyen egy adatlapja, külön fájlban pedig minden személynek szintén egy-egy adatlapja, amely a cégnéven vagy egy referenciaszámon keresztül kapcsolódik a cég adatlapjához.

Másik tipikus adatbázis valamely egyesület tagnyilvántartása. Ezek formája sem egyszerű, mert a tagok lehetnek egyének, házaspárok, családok is, ahol bizonyos információk közösek, mások egyéniek. Rendszerint ezeket is nehezen tudjuk kétszintű rendszer felépítése nélkül megoldani.

Bármilyen is legyen azonban az adatbázis szerkezete, mindig gondoskodni kell olyan rovatról, amely az adatlapokat egyedivé, a többitől megkülönböztethetővé teszi. A vezetéknév erre nem nagyon alkalmas. Ha van az adatlapon azonosításra alkalmas adat (például sorszám), akkor legjobb azt használni, ha pedig nincs, akkor magunknak kell „generálni" ilyet. Egyszintű, egyszerű adatbázisokban a telefonszám is betöltheti az azonosító kulcs szerepét, többszintű jegyzékben pedig kiegészítő jelöléssel kombinálható.

Az adatbevitel egyik fő követelménye a következetes pontosság, különösen azokban a rovatokban (adatmezőkben), amelyek szerint szortírozhatunk. Nem mindegy, hogyan használjuk az írásjeleket, teszünk-e pontot a dr. vagy egy rövidítés betűi után, mit írunk nagybetűvel, alternatív helyesírási lehetőség esetén melyiket használjuk (Mac vagy Mc), hogyan kezeljük az idegen ékezetes betűket stb.

Ugyancsak sok tévesztési lehetőséget kínál a telefonszámok írásmódja: hová teszünk zárójelet, melyik ponton mivel választjuk el egymástól a számcsoportokat (szóközzel vagy kötőjellel) stb. Ráadásul ezek használata nem is egységes. Adatbázisunk szempontjából is csak az a fontos, hogy amelyiket választjuk, ahhoz következetesek maradjunk.

A nevek írásával mindig sok a zűr. Ha adatbázisunkból csak címkéket nyomtatunk, akkor egyetlen rovatba tehetjük a vezetéknevet, a keresztnevet és a rangot. De akik ezt teszik, rendszerint megbánják, mert utána nem tudják adatbázisukat egyéni levelezésükhöz használni, a megszólítást egyedileg variálni. Célszerűbb tehát a névblokkot 3 vagy 4 rovatra bontva beírni, s még akkor is előfordulhat, hogy egyes címzettek a lehetséges sémákból kilógnak.

A címzést tartalmazó rovatok egyszerűbbek, de azért gondolni kell azok esztétikus elrendezésére a leveleken. Mind a nagyon hosszú, mind a nagyon rövid cím rossz benyomást kelt, ha nincs megfelelően elhelyezve. Külföldi levelezésnél eleve sokkal több információt kell tartalmaznia a címzésnek, az irányítószám külön ravatban való elhelyezése pedig bizonyos szortírozási-válogatási feladatok megoldásához nélkülözhetetlen.

Az adatbázisnak címkére többnyire nem kerülő, mégis fontos rovata a dátum, amely jelzi az adatok keletkezésének és legutolsó módosításának időpontját. Ebből ellenőrizhető, hogy megtörténik-e a rendszeres frissítés, minden adatbázis tényleges hasznosságának és értékének legfőbb forrása.

Ha egy meghatározott körhöz ugyanazt az információt akarjuk eljuttatni, akkor legracionálisabb a borítékra felragasztható címkék előállítása. Lehet persze közvetlen nyomtatáshoz előkészített borítékokat is vásárolni, vagy odaülhetünk a lézerprinterhez és etetgethetjük normál borítékkal... vagy a tehetősebbek még ennél is kényelmesebb önetetős berendezéseket használhatnak, azért általában mégis a címkék jelentik a legegyszerűbb megoldást.

Személyes megszólítású körlevelek esetén kézenfekvő alternatíva magára a levélre nyomtatni a címzést, és ablakos borítékot használni, aminek egyebek között az is az előnye, hogy mindenki a neki szóló levelet kapja, és nem valaki másét. Ehhez azonban levelezésünket adatbázis-kezelő rendszerben kell elkészítenünk, vagy be kell vinnünk a címeket a szövegszerkesztőbe. Leveleink és címanyagunk között az összeköttetést úgy is megteremthetjük, hogy adatbázis-kezelőnket és szövegszerkesztőnket párhuzamosan futtatjuk például DesqView vagy Windows alatt, és a Clipboardot használjuk összekötő kapocsként.

(Personal Computer World, 1991/12)

E számunk hirdetői

	Info#	Oldal
Cédrus (Diri)	30	31.
Cédrus (Foxpro 2.0)	14	36.
Cédrus (Irodabútor)	31	46.
Cédrus (Polaroid)	15	K IV.
Cédrus	6	29.
Cédrus (Staedtler)	13	33.
Cédrus (Áruház)	16	21.
Computerland	3	B IV.
Data Doctor (Vírus- és adatvédelem)	23	39.
DTP System (Calamus)	25	23.
Escom (Hardverinformációk)		Melléklet
Euro-Asia (Szaküzlet)	4	26.
Flag (Hardver/szoftver)	11	26.
Hewlett-Packard (Palmtop)	26	19.
IDG (PC World)	5	26.
Interag (Mitac)	1	B II.
Intram (Hardver)	8	68.
Koginform (Hardver)	28	50.
Kuoni (CeBIT '92)	19	34.
LSI (Könyvek)	9	25.
Macroda (Hardver)	6	42.
NTT 2000 (Irodatechnika)	7	42.
Qwerty (Hardver-szoftver)	24	36.
Realcomp (DTP)	10	27.
Rex Trade (Irodatechnika)	17	39.
Szoftver ABC (Szofver)	21	47.
Szolinfo (Számítógép-átalakítás)	20	47.
Toner (Canon)	22	47.
Távíró (Kommunikáció)	18	34.
Unitrade (Audio- és videokazetta)	12	36.
Unitrade (Hardver)	27	39.
Userland (Remind)	2	B III.

Van Önnek XT-je?

Szeretne inkább egy AT-t?

Számítógépét részegységek cseréjével már 16 450,– Ft-tól **átalakítjuk** 12 MHz-es AT-re!

Garanciával!

Ha Ön szereli, akkor csak 13 950,– Ft!

Fordulion hozzánk bizalommal!

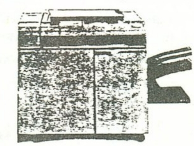
Szolinfo Kft.
Tel.: 173-6637
182-2646
166-5413

INFORMÁCIÓKÉRÉS: 20 ▼

Canon Canon

VÁLTSON SZÍNESRE Canon MÁSOLÓGÉPPEL!

CSÚCS AMIT TUD:



- Kicsinyítés
- Nagyítás
- Montírozás
- Tükörkép-készítés
- Képismétlés
- Poszter készítés
- 17 millió színáranyalat

Győződjön meg róla!

fénymásolás

TONER KFT
1095 Budapest, Mester utca 21.
Tel.: 113–1687, 134–3516

Canon Canon

INFORMÁCIÓKÉRÉS: 22 ▼

szoftver ABC Kft.

☏ : 201-6891
201-2011/131
🖷 : 201-8619
✉ : 1277 Budapest 23. Pf: 45.

Rövid határidővel szállított szoftvereink:

(Ár ÁFA-nélkül!)

DOSHun	6.000	Norton Backup for Windows	12.500
Ékszer	45.000	Norton Commander 3.0	13.000
Napló 2000	7.900	Norton Desktop for Windows	16.000
WinHun	6.000	Norton Editor	10.000
Adlib Pers. Music System	9.900	Norton Utilities 6.0	14.500
Adobe Type MGR Plus Pak	17.800	Novell Btrieve for Windows	57.000
Adobe TypeManager	10.000	Novell Netware 2.2 5-User	66.000
Aldus Pagemaker 4.0	67.000	Novell Netware 2.2 50-User	250.000
Ami Professional	40.950	Novell Netware 3.11 20-User	245.000
Anti Virus +	13.800	Novell Netware 3.11 100-User	489.000
Artline 2.0	49.800	Novell NetWare Lite	9.900
Carbon Copy Plus Host	23.500	Novell XQL	75.000
CC:Mail Fax View	110.000	Novell Xtrieve Plus	43.000
CC:Mail Gateway	134.000	Object Vision	19.000
CC:Mail Import/Export	108.000	On Target	32.000
CC:Mail link to UNIX Mail/uucp	75.000	On Track Disk Manager	7.000
CC:Mail Post Office Pak f. Windows	57.000	OrCad PCB Layout	192.000
CC:Mail Remote	34.000	OrCad VST	165.000
Charisma	39.000	PackRat V 3.0 for Windows	36.000
Checkit V3.0 /Hardware-Diagnos./	13.200	Paradox 3.5	40.000
Chiwriter Professionell 4.0	41.000	PC Anywhere IV	15.000
Clarion Profess. Developer	75.000	PC Globe	8.000
Clipper 5.01	75.000	PC Paintbrush IV Plus	18.000
Corel Draw 2.0	46.000	PC Tools 7.0	13.000
CP Anti-Virus	7.800	PC Tools 7.0 + MS DOS 5.0 UPDATE	18.500
Crosstalk for Windows	13.700	PerForm Pro for Windows	61.000
Dataperfect	32.000	Personal Rex	16.000
DBFast / Windows	32.000	PharLap 386 / VMM	26.000
DBXL 1.2d	34.000	PhotoStyler	94.000
Designer 3.1	46.500	PopDrop Plus	10.000
Deskview 386 2.4	22.000	Presentation Team 2.0	43.000
Desqview Qemm 386 V.6.0	10.500	Printer Assist	24.500
Desqview QRam 2.0	8.500	Printshop	7.000
Disk Optimizer	7.200	Procomm Plus	13.000
Draw Perfect	37.900	Publishers Paintbrush Windows 3.0	38.000
Draw Plus 1.1	20.500	Publishers Type Foundry	44.000
Easyflow 7.0	19.000	Q & A 4.0	34.000
F & A 4.0	48.000	Q Assist	19.000
FaceLift /Bitstream/ 13 Fonts	10.200	Quattro Pro 3.0	20.000
Facelift for Postscript	20.000	Quattro Pro SE	7.100
Fontasy 3.5	11.000	Quicksilver 1.3	42.000
Forest & Trees	46.000	R & R Clipper/Foxbase Modul	7.000
FoxPro 2.0	61.490	R & R Rel. Report Writer	19.000
FoxPro LAN 2.0	101.000	Reflex 2.0	22.500
FoxPro Toolbox 2.0	54.400	SCO Unix 3.2 Dev. Pack	97.000
Framework IV	64.000	SCO Unix 3.2 Oper. Sys.	84.000
Generic 3D Drafting	29.000	SCO Foxbase Plus 386	69.000
Go Script Plus 3.0	26.000	SCO TCP/IP Dev. Sys. for Unix 386	26.000
Grammatik IV for Windows	11.500	SCO Xenix 386 Oper. Sys.	73.000
Halo Windows Toolkit	54.000	Show Partner	11.000
Harvard Graphics 3.0	51.000	Show Partner Picture Pack	20.000
Harvard Project Manager III	53.000	Sideways	12.000
Hijaak 2.0	19.000	Sit Back for Windows	13.000
Intel LANShell	75.000	Smalltalk V	12.000
Intel LANSpool 386	87.000	Smalltalk V Windows 3.0	39.000
Intel LANSpool for LAN Manager	56.000	Smarterm 320	16.500
K-Edit 4.0	16.000	Software Bridge	12.000
LAN Assist Plus	29.000	Software Carousel 5.0	10.000
Landmark Speed Test 2.0.2	5.100	Sound Blaster	22.000
Laplink V 3.0	14.000	Source Print	12.000
Lotus 1-2-3 for Windows	52.000	SpeedStor	11.000
Map Assist	33.000	SPF/PC 2.1 Editor	20.500
MathCad 3.0 for MS Windows	45.000	SPSS/PC+	119.000
MathType for Windows	24.000	Stacker Harddisk Utility	13.000
Matrix Layout	22.000	Statgraphics 5.0	79.900
MS C Compiler 6.0	41.500	Superbase IV	53.000
MS DOS 5.0 Update	7.700	SuperCalc 5.0	41.000
MS Excel 3.0	43.000	SuperProject Expert	75.000
MS Flugsimulator Designer	5.500	Technobox CAD/2 for Windows 3.0	78.000
MS Fortran 5.1 PDS	41.500	Timeline 4.0	69.000
MS Macro Assembler PDS 6.0	16.000	Turbo Pascal for Windows 3.0	23.000
MS Pascal 4.0	24.000	Ventura Publisher Gold 3.0 WIN	88.000
MS Project for Windows	58.000	Vitamin C	36.000
MS Quick C for Windows	16.000	VM / 386 Multiuser	62.000
MS Visual Basic	17.000	WinConnect	11.000
MS Windows 3.0	12.000	Window Base	55.000
MS Windows Dev. Kit 3.0	42.000	Windows Maker Prof.	87.000
MS Windows Entertainment Pack	. 4.990	Windows Word for Word	10.000
MS Word 5.5	35.000	Winfax Pro	12.000
MS Word 5.5 Multispeller	11.000	Wingz for Windows	53.000
MS Word Exchange	7.200	Wordperfect 5.1	43.900
MS Word for Windows	43.000	Wordperfect Library	18.500
MS Word for Windows Multispeller	11.000	Wordperfect Office	18.500
MS Word for Xenix 386 / Unix 386	92.000	Wordstar 6.0	40.500
MS Works for Windows	22.000	XTree net Advanced	52.000
Nantucket Tools II 5.01	55.000	Zinc Interface Lib. 2.0 Borland	36.000
Netroom Single User	9.000	Zortech C++ Developers Ed. V3.0	49.500
NewsMaster II	8.000	Zortech C++ for Windows V 3.0	32.500
Norton Anti Virus	10.500	Zortech C++ Videokurs 6 x VHS/PAL	38.000
Norton Backup 1.2	14.000	Zortech C++ Views	44.000

Ami ide nem fért, azt is nálunk keresse!

INFORMÁCIÓKÉRÉS: 21 ▼

Szót érteni a Unixszal

Parancsolj velem!

Sorozatunk előző részében a Unix operációs rendszer felépítésével foglalkoztunk. Most azoknak a parancsoknak a részletes ismertetésére térünk ki, amelyek ismeretében alapfokon már használni lehet a rendszert. Feltételezzük, hogy az MS-DOS szolgáltatásait sokan ismerik, ezért — ahol lehetséges — megadjuk a Unix-parancsok DOS-megfelelőjét is.

Bejelentkezés a rendszerbe

Az előző részben leírtak szerint a Unix-parancsokat a shell értelmezi. Ahhoz, hogy a shellt használni tudjuk, kapcsolatba kell kerülni a Unix rendszerrel. A Unix szolgáltatásait csak azok a felhasználók érhetik el, akiket a rendszer azonosítani tud. Minden felhasználóhoz tartozik egy név és egy titkos jelszó. Bejelentkezéskor ezeket kell megadni a rendszernek. Alaphelyzetben általában a **login:** üzenet olvasható a terminálon. Itt a felhasználó nevét kell begépelni.

Ezután a begépelt névtől függetlenül megjelenik a **Password:** üzenet, amely után a titkos jelszót kell beütni (karakterei a titkosítás miatt nem jelennek meg a képernyőn). Ez az azonosítási rendszer az adatvédelmet szolgálja, célszerű nehezen kitalálható jelszavakat használni, és nem árt időnként újat választani. Vigyázni kell azonban arra, hogy az esetleg nagyon bonyolultra sikerült jelszavunkat ne felejtsük el. Nagy gond ilyenkor sincs, mert a rendszer adminisztrátora az elfelejtett jelszót meg tudja változtatni.

Ha nevünk és azonosítónk alapján a Unix felismert minket, a shell egy példánya elindul a terminálunkon. Mint tudjuk, a Unix alatt háromféle shell (a Bourne, a C és a Korn) használható. Ezek közül az igényünknek megfelelőt a rendszeradminisztrátor rendeli hozzá login környezetünkhöz. Ehhez a környezethez sok más jellemző is tartozik, például az ún. home könyvtár, amely a felhasználó fájljait tartalmazza, vagy a path, amelyen a fájlok keresése történik, vagy a prompt, amellyel a shell jelentkezik a képernyőn.

Alapértelmezésként a Bourne shell a '$' jelet írja a sor elejére. A C és a Korn shell üzenete egy sorszámmal kezdődik, amelyet a C shell esetén egy '%' jel, a Korn shell esetén pedig egy '$' jel követ. A sorszám értéke utasításonként eggyel nő. Ennek alapján a kiadott parancsok megkereshetők és újra végrehajthatók.

Tegyük fel ezek után, hogy bejelentkezésünk sikeres volt, és használhatjuk a rendszert. Természetesen szeretnénk minél gyorsabban parancsokat begépelni és látni ezek hatását.

Van egy parancs, amit az MS-DOS-ról nagyon keveset tudók is ismernek. Ha valaki számítógépet lát, szinte biztos, hogy ez lesz az első, amivel próbálkozni fog. Nem nehéz kitalálni, a dir parancsról van szó, amely a képernyőre írja az aktuális könyvtár tartalmát. Legyen a Unix alatt is ez az első feladat, amelyet megpróbálunk végrehajtani. Ehhez gépeljük be a következő utasítást: **ls -l**.

A parancsot később bővebben fogjuk ismertetni, elégedjünk meg most csak annyival, hogy hatására az aktuális könyvtár tartalma itt is a terminál képernyőjére kerül. Miután így elkezdtünk dolgozni a Unixszal, ismerjük meg részletesebben a parancsok használatát.

A Unix-parancsok felépítése

Egy Unix-parancs formája (szintaktikája) a következő:

```
parancs -opciók paraméterek
```

A Unix-parancsokat mindig kisbetűvel kell beírni. A parancsok paraméterei általában fájlok, könyvtárak, speciális fájlok. Vannak olyan parancsok, ahol van valamilyen alapértelmezésük, így megadásuk nem kötelező. Például az ls parancs, ha nem áll mögötte egy könyvtárnév, az aktuális könyvtárbeli fájlokat listázza.

A parancsok végrehajtását az opciók módosíthatják. Például az

```
ls -l
```

utasításban az -l opció adja meg, hogy a könyvtár elemeiről részletes információt kérek. Hiányában csak a fájlok neveit listázná a rendszer.

Általában, ha egy parancsot hibásan adunk meg (például valamilyen paraméter rossz vagy hiányzik), egy rövid ismertető jelenik meg a képernyőn. Ha például paraméter nélkül használjuk a remove parancsot, a következő üzenetet kapjuk:

Az ismertetett Unix-parancsok felsorolása

Típus	Unix	MS-DOS	Magyarázat
Könyvtárkezelő	pwd	cd	print working directory
	cd	cd	change directory
	ls	dir	list filenames
	mkdir	mkdir, md	make a directory
	rmdir	rmdir, rd	remove a directory
Fájlkezelő	cp	copy	copy file(s)
	mv	ren	move (rename) file(s)
	rm	del	remove file(s)

```
rm
usage: rm [-fir] file ...
```

Ez általában elég információ a parancs helyes használatához, ha azonban ennél többet akarunk tudni, a részletes ismertetést megtalálhatjuk a megfelelő kézikönyvben, illetve elolvashatjuk közvetlenül a képernyőről is. Ezt a man parancs teszi lehetővé. Használata:

```
man parancs_név
```

Célszerű a more utasítással együtt használni, amely lehetővé teszi, hogy képernyőoldalanként lépegessünk végig a dokumentumon. A parancs formája ilyenkor:

```
man parancs_név | more
```

Meg kell említenünk, hogy az olyan grafikus felhasználói felülettel rendelkező rendszerben, mint az SCO Open Desktop, létezik egy xman parancs is, amelynek hatására a dokumentum a nyomtatási képpel megegyező, jól olvasható formában jelenik meg a képernyőn.

A legegyszerűbb Unix-parancsok

A legegyszerűbb és egyben leggyakrabban használt Unix-parancsok a könyvtárakra és a fájlokra vonatkoznak. Felsorolásuk az 1.ábrán látható. A parancsok neve általában a működésüket leíró angol nyelvű szöveg rövidítése, ezért az angol leírást közöljük, amelynek alapján a parancsok jól memorizálhatók. Az ábrán az egyes parancsok MS-DOS-beli megfelelője is szerepel.

Könyvtárkezelő parancsok

Ha meg akarjuk tudni, hogy melyik az aktuális könyvtár, amelyben éppen dolgozunk, a pwd parancsot kell használni. Hatására a képernyőre íródik az aktuális könyvtár teljes pathneve. MS-DOS-megfelelője a cd parancs. Például:

```
pwd
/usr/gabor
```

Természetesen arra is kiváncsiak lehetünk, hogy mi az aktuális könyvtár tartalma. Ez a már ismert ls paranccsal listázható ki. A parancs önmagában csak a fájlnevek listáját adja meg. Bővebb információ a különböző opciók (kapcsolók) használatával kérhető. A leggyakrabban használt az -l (long) opció, amelynek hatására a fájlok nevén kívül sok egyéb, a fájlhoz tartozó információt is megkapunk. Az előzőek szerint MS-DOS alatt a dir parancsnak van hasonló hatása.

```
ls -l
```

Az ls -l parancs hatására megjelenő lista

```
drw-r--r--   1 gabor   group   46 Nov 13 13:51 bin
drw-r--r--   1 gabor   group   1283 Nov 13 13:51 src
-rw-r--r--   1 gabor   group   43 Nov 07 10:57 mailfile
```

Az első mező a fájlhozzáférési jogokat tartalmazza. A sor elején álló 'd' betű könyvtárat (directory) jelöl. A gabor és a group a felhasználói és csoportját azonosítja. Ezután a fájl hossza következik, majd a keletkezési ideje. Az utolsó oszlopban olvasható a fájl neve.

A Unix-ban is vannak ún. takart (hidden) fájlok, amelyek neve egy '.' karakterrel kezdődik, és amelyek a közönséges listázásban nem szerepelnek. Az ilyen fájlok az -a (all) opcióval jeleníthetők meg a könyvtár elemeinek listázásakor.

```
ls -la
```

Ha valamilyen más könyvtár tartalmát akarjuk megnézni, paraméterként megadhatjuk a kívánt könyvtár nevét. A parancs formája ilyenkor lehet például a következő:

```
ls -l /usr/lib/mail
```

Ha meg akarjuk változtatni az aktuális könyvtárat (egy másik könyvtárba akarunk kerülni), az MS-DOS-ból már ismert cd utasítást kell használni.

cd könyvtár_név

A könyvtár_név az új könyvtár pathnevét jelenti. Egy új könyvtárba el lehet jutni az aktuális katalógusból és a root (gyökér) könyvtárból is. Ha a cd parancsot a könyvtár_név argumentum nélkül adjuk ki, a saját ún. HOME könyvtárba kerülünk. Ezt a következő példa illusztrálja:

```
cd /usr/lib/mail
pwd
/usr/lib/mail
cd
pwd
/usr/gabor
```

Könyvtárak létrehozása és törlése az mkdir és az rmdir parancsokkal történik.

```
mkdir könyvtár_név
rmdir könyvtár_név
```

A parancsok MS-DOS alatt megszokott rövid formája, az md és rd itt nem használható. Figyeljünk arra, hogy a könyvtárak addig nem törölhetők, amíg tartalmaznak valamit.

Fájlokat kezelő utasítások

Most csak a legegyszerűbb, fájlokat manipuláló parancsokat ismertetjük. Segítségükkel fájlok másolhatók, törölhetők, illetve a tartalmuk megjeleníthető.

Másolásra két parancs is szolgál, a cp és az mv. Használatuk ugyanaz:

```
cp fájl_név1 fájl_név2
mv fájl_név1 fájl_név2
```

A cp (copy) eredményeként a fájl_név1 nevű fájlból létrejön még egy példány, amelynek fájl_név2 lesz a neve. Az mv (move) parancs végrehajtása után a fájl_név1 nevű fájl fájl_név2 néven fog szerepelni, azaz továbbra is csak egy példány lesz belőle. MS-DOS-beli megfelelője a ren (rename) parancs.

A fenti parancsok segítségével egy könyvtárba egyszerre több fájlt is bemásolhatunk:

```
cp fájl1 fájl2 fájl3 könyvtár
mv fájl1 fájl2 fájl3 könyvtár
```

A cp után a fájlok a célkönyvtárban is megjelennek, míg az mv hatására átkerülnek oda.

Könyvtárakat már tudunk törölni. Nézzük meg, hogyan lehet fájlokkal megtenni ugyanezt. Az MS-DOS alatt erre a del parancs szolgál, míg a Unixban az rm (remove) utasítást kell használni. Paramétereként több fájl neve is szerepelhet. Az utasítás formája:

```
rm fájl_név
rm fájl1 fájl2 fájl3
```

A fájlok tartalmának megjelenítésére a cat parancsot használjuk. Az előző parancsokhoz hasonlóan paraméterként itt is megadható több fájl neve. MS-DOS-megfelelője a type utasítás.

```
cat fájl_név
cat fájl1 fájl2 fájl3
```

Ha másképpen nem rendelkezünk, a fájlok tartalma az stdout-ra (standard output), azaz a terminál képernyőjére

kerül. Lehetőség van azonban arra is, hogy a listázást egy fájlba irányítsuk. Ezt a '>' operátor segítségével tehetjük meg. A parancs formája ilyenkor a következő:

```
cat fájl1 fájl2 fájl3 > eredmény
```

Végrehajtása során létrejön az eredményfájl, amelybe fájl1, fájl2 és fájl3 tartalma kerül. Ha már létezett az eredmény nevű fájl, akkor írása előtt a régi tartalma megsemmisül. Ha azt akarjuk, hogy az is megmaradjon, a '>' operátort kell használni. Ilyenkor a paraméterként felsorolt fájlok az eredményfájl végére másolódnak.

Fájlok megjelenítésére használható még a more parancs. Ez két formában alkalmazható, önmagában és szűrőként. Mind a két esetben a képernyőre írás 25 soronként megszakad, és az Enter billentyű leütése után folytatódik. Segítségével hosszabb fájlok, könyvtárak stb. is gond nélkül végigolvashatók. Használata a következő:

```
more fájl_név
```

Szűrőként egy másik kiíró paranccsal együtt alkalmazható. A Unix egyik jól használható eszköze az \n. pipe vagy cső. Egy parancs kimenetét köti össze egy másik parancs bemenetével, azaz az első parancs végrehajtása során keletkező kimenőadatok közvetlenül a másik parancs bemenetére kerülnek. Nincs szükség közbenső fájlokra, kevesebb a hibalehetőség. A két parancs egy időben is dolgozhat, azaz ahogy a cső végén megjelenik egy adat, elkezdődhet a további feldolgozása. Ezzel a módszerrel több parancs is összefűzhető. Végezetül néhány példa a more utasítás pipe-os használatára.

```
cat fájl_név | more
ls -l | more
ls könyvtár_név | more
man utasítás_név | more
```

Déri Gábor

UFF!

A Unix operációs rendszer hazai térhódításával egyre nagyobb számban vannak azok a Unix-felhasználók, akiket mind szélesebb körben kell megismertetni a nyílt rendszerek világával. A Unix-rendszerintegrátori feladatokat magára vállaló MemoLuX cég tapasztalatai alapján célszerűnek tartja, hogy a Unix-felhasználókat valamilyen formában „összefogja".

A nyílt rendszerek hatékonyabb terjedését segítheti, ha a felhasználók minél jobban tájékozottabbak az egyes területeken megvalósított applikációkról. Ehhez azonban olyan fórumra van szükség, ahol a felhasználók nemcsak a fejlesztőkkel és kereskedőkkel találkozhatnak, hanem egymás közti kapcsolatteremtésre is lehetőségük nyílik. A MemoLuX Kft. januárban induló rendezvénysorozata ezt a kettős célt igyekszik megvalósítani.

A most útjára induló Unix Felhasználók Fórumának (UFF) szakmai színvonalát jelzi az a tény, hogy a rendezvénysorozatot a Mérés és Automatizálási Tudományos Egyesület (MATE) Számítógép Szakosztálya felvette hivatalos szakmai programjába. Az újonnan induló „mozgalmat" az Alaplap — mint PC Turbo Klubunk testvérklubját — támogatja, így olvasóink is részt vehetnek a rendezvényeken.

Az UFF első összejövetelén „beléphetünk" az Open System világába. A nyitó rendezvény témájául a szervezők a PC-s Unix témakörét választották. Ugyanis valamennyi Unix operációs rendszerre áttérő felhasználót érdekli — sőt érzékenyen érinti — a DOS—Unix átmenet zökkenőmentes megvalósítása, ennek konkrét hazai megvalósítása szerepel a a fórumon. De az összejövetelen szó lesz a PC-k hálózatba integrálásának módjai mellett a helyi hálózatokkal való összekapcsolás lehetőségeiről is. A rendezvény befejezéseként pedig röviden ismertetik a negyedik generációs adatbázis-kezelők legfontosabb jellemzőit, s ami a felhasználók számára talán a legfontosabb — néhány konkrét alkalmazási területről is beszámolnak.

Az újszerű, s főleg rendkívül hasznos UFF-rendezvényekről, azok időpontjáról és témájáról, a továbbiakban rendszeresen tájékoztatjuk az Alaplap olvasóit. A tervek szerint kéthavonta lennének UFF-összejövetelek, amelyek teljesen nyilvános és ingyenes szakmai rendezvények lesznek. Az összejöveteleknek a Cédrus Rt. Karolina út 17. alatti újonnan megnyílt Konferenciaterme ad otthont, ahová — a meghívott felhasználók mellett — természetesen az Alaplap nyílt rendszerek iránt érdeklődő valamennyi olvasóját is várjuk.

Az UFF első összejövetelére 1992. január 22-én 15 órakor kerül sor.

Sziebig Andrea

Elöljáróban

A Unix operációs rendszer egyre gyorsabb tempóban terjed a világon, aminek egyik oka az, hogy számos szolgáltatása (parancsa) van, amelyek sokféle módon kombinálhatók egymással. Megismerésének éppen ez az egyik nehézsége: sok a parancs, nem könnyű egy adott feladathoz megtalálni a legjobb eszközöket.

Cikksorozatunkban a Unix néhány szolgáltatására szeretnénk felhívni olvasóinknak a figyelmét, akik a legalapvetőbb ismeretekkel már rendelkeznek erről az operációs rendszerről. (Felhívjuk a figyelmet az Alapjárat rovatban néhány hónapja rendszeresen megjelenő cikkekre is, melyek a Unix operációs rendszer ismertetését alapfokon kezdték.

Részletesebben szólunk a parancsértelmező (shell) programozásáról azért, mert a rendelkezésre álló irodalomból a szintaktikai szabályok csak eléggé fáradságosan bányászhatók ki, viszont nagyon hatásos eszközt kap a kezébe a felhasználó, ha van egy kis türelme a tanuláshoz.

A C nyelv Unix-felületéről is szó lesz a későbbiekben. Ebben a témakörben a ki- és beviteli utasításokkal bővebben foglalkozunk — egyrészt azért, mert ezek alapvető fontosságúak, másrészt azért, mert itt is van néhány hasznos, ám a kézikönyvekben túl tömören leírt tudnivaló.

Sorozatunk a shellprogramozással indul — ez néhány hónapig eltart majd. Utána hozzuk a C-s részeket, ezeket követi a Unix hatékonyságvizsgáló eszközeinek az ismertetése, majd a Berkeley-bővítések, a folyamatkommunikáció; s hogy maradjon még valami meglepetés is, a felsorolást itt abbahagyjuk...

A Unix shell programozása I.

Leggyakrabban, legegyszerűbben

A Unix parancsértelmezője egy magas szintű programnyelv, amely számos szolgáltatást nyújt, de a jelölésmódja nagyon tömör — ami a használatnál előny, alapos megismerését viszont fáradságossá teszi. Az ismerkedést az is nehezíti, hogy metakarakter-készlete nem egyezik meg a szövegfeldolgozó programok (ed, sed, grep, awk) készletével, a szintaktikája pedig számos ponton hasonlít a C nyelvére, noha kis eltérések ilyenkor is vannak.

A shellprogramozás előnyei

Kezdőknek nehézséget szokott okozni a paraméterátadás technikájának elsajátítása is. (Néha meg kell akadályozni, hogy a shell a metakaraktereket kifejtse — ezeket változatlanul át kell adnia az általa elindított programnak.) A shell program egyszerű ASCII szövegfájl, ezért könnyű létrehozni, megtalálni, megérteni és karbantartani. Nem kell fordítani és szerkeszteni, a program nagyon gyorsan létrehozható, belőhető, módosítható és törölhető. Kevesebb helyet foglal el a háttértárolón, mint egy futtatható bináris modul. Ha a sebesség nem kritikus, számos esetben előnyösebb shell programot fejleszteni, mint például C-ben megírni azt. A C programok létrehozásához is komoly segítséget jelent, ha először shellben írjuk meg a programot, mert ez nagyon gyorsan megtehető, majd a funkcionális hitelesítés után hozzáláthatunk az átíráshoz. Nem véletlen, hogy a rendszerprogramok jelentős hányada shell script!

A következőkben bemutatjuk a shell programozásához szükséges főbb ismereteket. Az elmélyülést azzal kívánjuk könnyíteni, hogy először a leggyakrabban előforduló és legegyszerűbb dolgokról lesz szó, illetve az összetettebbeknek csak egy részét említjük meg. A precízebb (és ezért kezdőknek egy kicsit nehezebben érthető) leírás később következik.

Parancsfájlok létrehozása

A parancsokat begépelhetjük terminálról, vagy elhelyezhetjük egy parancsfájlban (ezeket hívják shell scripteknek). A parancsfájlt a létrehozása után végrehajthatóvá kell tenni a chmod utasítással:

```
chmod +x comfile
```

Ezután a parancs elindítása a fájl nevének leírásából áll:

```
comfile
```

A parancsokat külön sorban kell elhelyezni; ha egy sorba több parancsot akarunk írni, akkor azokat pontosvesszővel kell elválasztani egymástól. Egy egyszerű parancs a nevéből és a paramétereiből áll, ezeket tetszőleges számú szóköz és tabulátor választhatja el egymástól. Ha egy parancs és a hozzá tartozó paraméterek nem férnek el egy sorban, akkor a sorvégjel elé egy \ jelet kell tenni. Például:

```
cat XX YY \
ZZ
```

Ennek a parancsnak 3 paramétere van; hatására kilistázódik a képernyőre az XX, YY és ZZ fájl tartalma.

Az echo utasítás

Az echo utasítás a paramétereit egy-egy szóközzel elválasztva kiírja a standard kimenetre. Például:

```
Parancs:
echo   AAAA                BBBBB
Kimenet:
AAAA BBBBB
```

Ez az utasítás az üzenetek kiíratásán kívül a belövéshez is jól használható, gyorsan ellenőrizni tudjuk a segítségével a paraméterek és változók értékét. Ezenkívül minden olyan esetben az echóra van szükségünk, amikor a standard kimenetre írni akarunk.

Pozicionális paraméterek

A parancssorban átadott paramétereket (pozicionális paramétereket) $1, $2 stb. néven tudjuk elérni:

```
comfile tartalma:
echo $1 $2 $3
Parancs:
comfile AAA  BB CCCCC
Kimenet:
AAA BB CCCCC
```

A nem létező pozicionális paramétereket a shell nullstringgel helyettesíti. Ha például a fenti parancsfájlt így hívjuk meg:

```
comfile A B
akkor a kimenet a következő lesz:
A B
```

A $0 pozicionális paraméter mindig az elindított program nevét jelenti:

```
comfile tartalma:
echo $0 : Hiba!!
Parancs:
comfile
Kimenet:
comfile : Hiba!!
```

A $0 paraméter hibaüzeneteknél azért előnyös, mert a program nevének megváltoztatásakor az üzenő részt nem kell módosítani. A $1, $2 stb. jelöléssel csak az első 9 paramétert tudjuk elérni. A shift utasítás hatására minden paraméter ($0 kivételével) „balra" lép: Az eddigi $1 eltűnik, $1 értéke az eddigi $2-ével lesz egyenlő. Például:

```
comfile tartalma:
echo $1 $2 $3 $4 $5 $6 $7 $8 $9
shift
echo $1 $2 $3 $4 $5 $6 $7 $8 $9
Parancs:
comfile 11 22 33 44 55 66 77 88 99 00
Kimenet:
11 22 33 44 55 66 77 88 99
22 33 44 55 66 77 88 99 00
```

Az összes pozicionális paramétert ($0 kivételével) a $* jelöléssel tudjuk elérni, a számuktól függetlenül. Például:

```
comfile tartalma:
echo $*
Parancs:
comfile 11 22 33 44 55 66 77 88 99 00 aa bb
Kimenet:
11 22 33 44 55 66 77 88 99 00 aa bb
```

A pozicionális paraméterek számát a $# nevű shellváltozó tartalmazza:

```
comfile tartalma:
echo $#
Parancs:
comfile 11 22 33
Kimenet:
3
```

Shellváltozók

A shell változóit nem kell deklarálni. Stringeket tartalmaznak, értéküket ugyanúgy használhatjuk, mint az explicit módon megadott stringeket. A változónevek betűvel vagy aláhúzással kezdődő, betűből, számjegyből és aláhúzásból álló stringek:

```
X=ABC
echo $X DEF
Kimenet:
ABC DEF
```

A példából láthatjuk, hogy a shellváltozók tartalmára a $név jelöléssel hivatkozhatunk. Fontos megfigyelnivaló ezenkívül, hogy az értékadásnál az egyenlőségjel egyik oldalán sem szerepelhet szóköz. A shell így ismeri fel, hogy változó értékadásáról van szó. Ha így írjuk:

```
X =ABC
```

akkor a következő hibaüzenetet kapjuk:

```
comfile: X: Not found
```

A shell ekkor úgy gondolta, hogy egy X nevű parancsot akartunk elindítani =ABC paraméterrel. Ha az egyenlőségjel másik oldalára teszünk szóközt:

```
X= ABC
akkor ezt az üzenetet kapjuk:
comfile: ABC: Not found
```

Most a shell úgy értékelte, hogy az X nevű változónak nullstring értéket akartunk adni, majd az ABC nevű parancsot szerettük volna elindítani. Éles szemű olvasók azonnal felkiálthatnak: hát nem egy parancsot lehet csak egy sorba írni? A válasz az, hogy az értékadás ebből a szempontból kivétel. Ennek a szabálynak az alkalmazására később még visszatérünk.

Ha olyan stringet szeretnénk egy változóba betölteni, amely szóközöket és/vagy tabulátorokat is tartalmaz, akkor idézőjelet kell használnunk:

```
X=„Mi a parancs?"
```

echo $X

```
Kimenet:
Mi a parancs?
```

Az idézőjel helyett aposztrófot is használhatunk:

```
X='Mi a parancs?'
```

A két megoldás közötti különbségről később lesz szó.

Programok elindítása a háttérben

Ha a parancs és a paraméterei után & jelet teszünk, akkor a rendszer a háttérben fogja azt elindítani, a terminálnál közben visszakapjuk a vezérlést:

```
command &
```

Programok leállítása

Interaktívan futó programokat interrupttal lehet félbeszakítani. Az, hogy milyen billentyűhöz van kötve a megszakításjel beadása, az stty paranccsal állítható be. A háttérben futó programokat a kill paranccsal tudja megszakítani a program tulajdonosa vagy a superuser (root):

```
kill 2341
```

A kill parancs paramétere a megszakítani kívánt folyamat azonosítója (process id-je). Ezt a ps utasítás segítségével tudjuk megkérdezni a rendszertől.

Megjegyzések shell scriptekben

A # jeltől kezdve a sor végéig a shell egy commentnek tekinti a sort:

```
echo XXXXXXX # ez egy comment
```

Az üres utasítás

A : üres utasítást jelent, az utána írt szöveget paramétereknek tekinti a shell, és megpróbálja kifejteni — ami vezethet futási hibára. A # bevezetése óta az üres utasítás elvesztette a jelentőségét.

Átirányítás

A programok standard bemenetét és kimenetét nagyon egyszerűen át lehet irányítani, ez a Unix egyik olyan sajátossága, ami lehetővé teszi a meglévő építőkockákból való építkezést. A kimenetnek fájlba való átirányítására szolgál a > és a >> jel. A

```
command > FILE
```

parancs hatására két dolog történik:

1. Ha a fájlhozzáférési jogok lehetővé teszik, létrejön a FILE nevű fájl, vagy ha már létezett ilyen, akkor a tartalma elvész, a hossza 0-ra redukálódik.

2. A command nevű program által a standard kimenetre küldött adatok bekerülnek a FILE nevű fájlba. Például a

```
date > LOG
```

parancs hatására a LOG nevű fájl tartalma a következő lesz:

```
01:04:34 PM Fri 13 Sep 1991 MEZ
```

A következő parancs hatására a command nevű program által a standard kimenetre küldött adatok a FILE nevű fájl végéhez hozzáfűződnek; ha FILE eddig nem létezett, akkor a rendszer létrehozza:

```
command > FILE
```

Az átirányítást vezérlő rész a parancs elé is kerülhet:

```
> FILE command
>  FILE command
```

Egy FILE nevű üres fájl előállításához, illetve megléte esetén 0 hosszra csonkításához elegendő a következő parancs:

```
> FILE
```

Írás a standardhiba-kimenetre

A standardhiba-kimenetre (a 2-es fájldeszkriptorra) a shell scriptből a következő módon tudunk írni:

```
echo hibal  >&2
cat hibafile >&2
```

A > és a & jel között nem szabad szóközt hagyni. (Hiszen elvileg lehet egy &2 nevű fájlunk, amelybe át akarhatjuk irányítani a parancs kimenetét! A Unixban nemcsak betűkből és számjegyekből állhatnak a fájlnevek.)

A standardhiba-kimenet (stderr)

A programok által a standard error kimenetre írt adatokat a következő módon lehet egy fájlba irányítani:

```
command 2> FILE vagy command 2> FILE
```

A 2-es szám és a > jel között nem szabad szóköznek lennie! (Ekkor a shell úgy értelmezné, hogy a command nevű parancsot akarjuk elindítani „2" paraméterrel.) Példa az átirányításra:

```
comfile tartalma:
echo ez megy a standard kimenetre
echo EZ MEGY A STANDARD ERROR
KIMENETRE >&2
Parancs:
comfile > F1 2>F2
F1 tartalma:
ez megy a standard kimenetre
F2 tartalma:
EZ MEGY A STANDARD ERROR KIMENETRE
```

Az stderr átirányítása a standard kimenetre

Erre akkor van szükség, amikor a standard error kimenetet bele akarjuk irányítani egy pipe-ba, például azt akarjuk, hogy egy nyomtatási parancs által küldött hibaüzenet a nyomtatón jelenjen meg. Az eljárás a következő:

```
comm 2&1
```

A standard bemenet átirányítása fájlra

A következő parancs hatására a comm nevű program a terminál helyett a FILE nevű fájlból fog olvasni, feltéve, hogy az létezik:

```
comm < FILE
```

A /dev/null fájl használata

A /dev/null fájl a nullázó eszköz a Unix rendszerben; a beleírt adat eltűnik, illetve ha a tartalmát belemásoljuk egy fájlba, akkor az 0 hosszra csonkítódik. Akkor szoktunk ebbe a fájlba írni, ha azt akarjuk, hogy egy program ne írjon üzeneteket a képernyőre:

```
comm 2> /dev/null
```

Erre részben akkor van szükség, amikor magunk akarjuk a képernyőt kezelni, részben akkor, amikor nem akarjuk egy időmérés eredményét a kiírás idejével „meghamisítani". Nulla hosszúságú fájlt így is szoktak létrehozni:

```
cp /dev/null FILE
cat /dev/null > FILE
```

Pipe-ok használata

A pipe egy átmeneti tároló, egy memóriában megvalósított FIFO, aminek a segítségével szűrő jellegű programok kaszkádba köthetők. (Szűrőnek azokat a programokat szokták nevezni, amelyek a standard bemenetükről olvasnak, és a standard kimenetükre írnak.) Lényeges, hogy memóriában van megvalósítva, mert így gyors a pipe-on keresztül végzett kommunikáció a programok között, ráadásul a többi program működését is kevéssé lassítja, mert nem hoz be diszkműveleteket. A pipe jele: |. Ha például meg akarjuk számolni, hogy hány bejegyzés van az aktuális munkakatalógusunkban, akkor ezt így tehetjük meg:

```
ls | wc -l
```

(A wc utasítás az -l opcióval a bemenet sorait számolja meg.) A csővezetékek megcsapolhatók a tee nevű parancs segítségével, amelynek paraméterként azt a fájlnevet kell megadni, ahová a kimenetet irányítani akarjuk:

```
ls | tee F1 | wc -l
```

Ekkor az F1 nevű fájlban az ls parancs kimenete fog megjelenni.

A standard kimenet átirányítása a parancssorba
(parancsbehelyettesítés, command substitution)

Rendkívül jól használható a shellnek az a szolgáltatása, hogy egy parancs kimenetét vissza lehet irányítani magába a programba az ' aposztróf használatával. Például:

```
comfile tartalma:
X='command2'
echo $X
command2 tartalma:
echo 11 22 33 44
echo 55 66 77 88
Kimenet:
11 22 33 44 55 66 77 88
```

A " jelek közé írt parancs standard kimenetéből a shell egyetlen stringet készít úgy, hogy a sorvégjeleket szóközre cseréli, és ezt a stringet behelyettesíti a " közé írt parancs helyére. Ha például készítünk egy savelist nevű fájlba egy listát a menteni kívánt fájljainkról, akkor nem kell leírnunk mentéskor a fájlneveket:

```
tar cv4 file1 file2 file46
```

hanem így is eljárhatunk:

```
tar cv4 'cat savelist'
```

Így biztosan nem fogunk egy névről sem megfeledkezni. (A tar parancs a floppyra vagy a mágnesszalagra való mentés eszköze.) Ha be akarjuk tölteni egy változóba az éppen aktív felhasználók számát, akkor ezt kell írnunk:

```
X='who | wc -l'
```

Ha egy változóba az aktuális munkakatalógus nevét szeretnénk elhelyezni:

```
Y='pwd'
```

A parancsbehelyettesítés tulajdonképpen az eljáráshívás általánosítása. Ha például egy C programban egy változóba be akarunk tölteni egy függvény által visszaadott adatot, ezt meg tudjuk tenni:

```
X = func();
```

A C-ben nincs arra mód, hogy egy elindított program kimenőadatait visszatöltsük közvetlenül egy változóba; a shellben ezt a feladatot látja el a parancsbehelyettesítés. Előfordul, hogy a létrejövő lista a shell belső tárolójának méretét meghaladja, ekkor „Arg list too long" hibaüzenetet kapunk. Ilyenkor több részre kell osztani a listát.

Olvasás a standard bemenetről

A read v1 v2 ... utasítás hatására a shell beolvas egy sort a standard bemenetről, az első szót a v1 változóba, a másodikat a v2 változóba stb. tölti be. Ha több szó van a sorban, mint ahány változónevet megadtunk, akkor a sor fennmaradó részét az utolsó változóba teszi. Például:

```
comfile tartalma:
read X rest
echo $X
echo $rest
Parancs:
comfile
```

A terminálról ezt írjuk be:

```
111 22 33 44 55
Kimenet:
111
22 33 44 55
```

A line utasítás hatására a shell beolvas egy sort a standard bemenetről, és kiírja azt a standard kimenetre. Ha a beolvasott sort egy változóba akarjuk betölteni, akkor ' jeleket kell alkalmaznunk:

```
comfile tartalma:
X='line'
echo $X
Parancs:
comfile
```

A terminálról ezt írjuk be:

```
aaaa bb cc ddd
Kimenet:
aaaa bb cc ddd
```

Stringek összefűzése

A változókban és a pozicionális paraméterekben lévő stringeket literálisan megadott stringekkel nagyon egyszerűen össze lehet fűzni, csak egymás mellé kell írni őket:

```
comfile tartalma:
X=aaa/
Y=bbb
Z=$X$Y/$1/ddd
echo $Z
Parancs:
comfile ccc
Kimenet:
aaa/bbb/ccc/ddd
```

Rendszerint teljes fájlnevek előállítására használják ezt a műveletet. Ha egyértelművé kell tenni, hogy az egyes részeknek hol van a határuk, akkor kapcsos zárójelet kell alkalmaznunk:

```
X=aaa
Y=${X}bbb
echo $Y
Kimenet:
aaabbb
```

Ha nem használunk zárójelet, akkor a shell az Y változóba egy Xbbb nevű változó tartalmát tölti be (tehát egy nullstringet).

Nemes Mihály

Modula-2

Most mutasd meg!

Programjainkban az egyes értékeket változókban tároljuk. A forráskódban a változók azonosítóival tulajdonképpen a változó által elfoglalt memóriaterületre hivatkozunk. Az adott változó típusától függ az elfoglalt memóriaterület mérete. Az általunk deklarált változóazonosítók és az aktuális memóriaterület közötti megfeleltetést a fordító végzi el.

A Modula-2-ben, mint minden „tisztességes", rendszerprogramozásra alkalmas nyelvben, lehetőség van a változók címeinek kezelésére, valamint futásidőben (tehát nem a fordító, hanem a program által) létrehozott változók manipulálására.

A futásidőben létrehozott — ún. dinamikus — változók használata programjainkban lehetővé teszi a memória optimális kihasználását, az aktuális igényekhez igazodó megoldások elkészítését, komplex, illetve dinamikusan változó struktúrájú adattípusok létrehozását, és még sok minden más, az igazán professzionális programokat jellemző megoldást. Mint látni fogjuk, a memóriacímek kezelése és a dinamikus változók használata szorosan összefüggő programozástechnikai témakörök, ezért természetes, hogy együtt tárgyaljuk őket.

Bevezetőként egy kis példa. Tegyük fel, hogy egy kis személyügyi nyilvántartással kapcsolatos programot kell elkészítenünk. Természetesen a személyi adatot rekordtípussal írjuk le mondjuk az alábbi módon:

```
TYPE
  PersonInfo = RECORD
    name  : ARRAY [0..16] OF CHAR;
    age   : [0..125] ;
    shift : (day, night) ;
    info  : ARRAY [0..100] OF CHAR ;
  END (* RECORD PersonInfo *) ;
```

Ezek után lehetőségünk van változók deklarálására:

```
VAR
  john, mary, peter, david :
PersonInfo ;
```

Persze ez csak abban az esetben járható út, ha előre tudjuk a nyilvántartásba veendő személyek számát (és nevét), ami természetesen a lehető legritkább esetben fordul elő a mindennapi gyakorlatban. Kicsivel jobb a helyzet, ha tömböt deklarálunk az adatok nyilvántartására:

```
VAR
  persons : ARRAY [1..100] OF
PersonInfo ;
```

Ebben az esetben a programban előre 100 személy adatainak nyilvántartására foglalunk helyet, holott egy tipikus alkalmazásban átlagban csak 20-30 személyt tartunk nyilván. Vagyis programunk feleslegesen köti le a számítógép memóriáját, azaz az egyik legfontosabb erőforrását.

De van itt egy másik probléma is. Mi történik akkor, ha programunkat egy olyan cég is alkalmazni kívánja, ahol 101 munkatárs dolgozik? Természetesen átírjuk a programunkat úgy, hogy a lefoglalt terület most már ezt az igényt is kielégítse, hiszen ha megfogadta a programozó a tanácsomat, akkor a forráskódban mindössze egyetlen konstans értékét kell megváltoztatni, és utána újra fordítani az egészet. Ez mind nagyon szép, de mi van akkor, ha a program — hibái ellenére — nagy siker lesz, és a legkülönbözőbb cégek szeretnék megvásárolni. A cégek között azonban olyanok is vannak, melyek 5, és olyanok is, amelyek 50 000 dolgozóval számolnak. Nyilvánvaló, hogy a cégenkénti forráskód-karbantartás, különösen a módosítások nyilvántartása, megoldhatatlan feladat.

Ahhoz, hogy a statikus változók rugalmatlanságaiból adódó hátrányokat (fix elhelyezkedés, méret és struktúra) kiküszöböljük, a programokban dinamikus változókat használhatunk, melyek elhelyezkedését, méretét és struktúráját magunk vagy a futási környezet határozhatja meg. Természetesen az ilyen általunk létrehozott dinamikus változókra már nem hivatkozhatunk azonosítókkal a forráskódban, hiszen a program futása során az azonosítók szerepét már a fordító és a programtöltő által meghatározott címek vették át. Ha már azonosítót nem használhatunk, magától értetődik az az igény, hogy legalább a dinamikus változók címeire tudjunk a programunkban azonosítóval

Névjegy

Mutató

A professzionális programozók számára a Modula és más programozási nyelvek által nyújtott eszközök közül a mutató típusok a legfontosabbak. A mutató típusok segítségével valósítható meg a komplex adattípusok, valamint rugalmas és dinamikus adatszerkezetek létrehozása és kezelése. A mutatók más változókra történő hivatkozásra használatosak. A mutatónak a hivatkozott változó létrehozása előtt már léteznie kell, és csak a hivatkozott változó megszűnése után szűnhet meg. Ebből adódóan a mutatók tipikusan, de nem kizárólag statikus változók. A mutatók segítségével lehetségessé válik a program futása során dinamikusan létrehozott változók kezelése. A Modula-2-ben nincsen előre definiált mutató típus. A mutatók eltérő típusú dinamikus változók kezelésére szolgálnak, ezért a referált típust a deklarációban meg kell adni. A hivatkozott típus megadásával a típusellenőrzés a mutatókra is kiterjed, ami a programozót durva hibák elkövetésétől óvja meg. Természetesen a típusátalakítások a mutatókon és a dinamikus változókon is elvégezhetőek, így ez nem jelent gondot olyan esetekben, amikor szándékosan akarjuk a típuskonvenciókat megsérteni.

hivatkozni. Erre a célra a Modula-2-ben a mutató típusok állnak a rendelkezésünkre. Ahogy arra a mutató elnevezés is utal, ezeknek a változóknak a szerepe a dinamikusan (vagy akár statikusan) létrehozott változók referenciájának/hivatkozásának, azaz a fizikai címének tárolása. A fizikai cím segítségével aztán már könnyedén tudjuk a dinamikus változókat is kezelni, mivel a mutatók segítségével hivatkozhatunk egy-egy dinamikusan létrehozott objektumra. Másképpen fogalmazva: a mutató típusú változók által felvett értékek fizikai címek lehetnek. Egy mutató a program futása során akár több objektumra is mutathat (természetesen nem azonos időpillanatban).

A mutatókkal való programozás a kezdők számára általában gondot szokott okozni, mivel a programozástechnikai elmélet igyekszik távol tartani magát a „hardver valóságától", és jobban szereti az absztrakció világát. A mutatókkal kapcsolatban persze elkerülhetetlen a memóriacímek megemlítése, de nem igazán rögzül az a tény, hogy az addig használt változónevek valójában fizikai tárcímek és nem karakterláncok. Sokkal jobb helyzetben vannak ilyen szempontból azok a programozók, akiknek assembly programozói hátterük van, mivel számukra a program absztrakt és fizikai megjelenése közötti megfeleltetés világosabb, nekik viszont az elmélet más részeivel kell megküzdeniük. Az ideális rendszerprogramozó természetesen mindkét megközelítési módot ismeri, és megfelelően használni is tudja. Sajnos ez a szakemberképzésben nem mindig szempont.

Mint az a definícióból kitűnik (lásd keretben), a mutató maga lehet statikus vagy dinamikus, de minden dinamikus hivatkozási sor elején (már ami a hivatkozási irányt illeti) egy statikus mutató található (ha végiggondoljuk, ez magától értetődik), tehát semmi okunk programunkban kizárólagosan dinamikus változókat használni. Ez már annál is inkább igaz, mivel a mutatóváltozók helyfoglalása sok alaptípus helyfoglalásánál nagyobb (méretoptimalizálás), illetve a futásidőben történő hivatkozás kielégítése időt vesz igénybe szemben a statikus hivatkozásával (futásidő-optimalizálás). Természetesen a tömböket sem kell száműzni programjainkból, csupán a feladatot kell végiggondolnunk úgy, hogy el tudjuk dönteni: a részfeladatokat milyen adattípusokkal, illetve statikus vagy dinamikus változókkal célszerű megoldani (rendszertervezés, megoldásoptimalizálás).

Ezeknek a kérdéseknek az eldöntése adja a programozás savát-borsát. (A kódolás ehhez képest sok esetben unalmas rutinmunkának tűnik.)

Most, hogy már láttuk vagy legalábbis utaltunk a dinamikus típusok hasznosságára, itt az ideje megnézni, hogyan is hozhatjuk őket létre. A Modulában — mint sok minden más — a memóriakezeléssel kapcsolatos eljárások is külső könyvtármodulban találhatóak, tehát nem a nyelv elemei. A Modula-2 tárkezelő eljárásai a „Storage" modulban találhatók. Dinamikus változókat az ALLOCATE eljárás segítségével hozhatunk létre. Az eljárás két paramétere a létrehozott változóra hivatkozó pointer, amelyben a létrehozott változó memóriabeli címét — azaz a memóriamenedzser által a változó számára allokált/lefoglalt terület kezdőcímét — tároljuk a későbbi hivatkozások számára, illetve az allokálandó terület mérete, ami általában (de nem mindig) a hivatkozott típus által elfoglalt terület méretével egyezik meg. Például ALLOCATE (pointer, SIZE(T)), ahol pointer egy T típusú változóra hivatkozó mutató. Az, hogy a méretparaméter tetszőlegesen megválasztható, sok trükkös megoldást tesz lehetővé. Persze mint minden olyan esetben, amikor elhárítjuk a fordító és a futásidejű rendszer óvó kezeit, itt is sok a hibalehetőség. Egy profit persze a nehézségek nem rettenthetnek vissza. A Modula-2

Pontosan és szépen

Definíció:

P mutató típusú változók egy másik T típusú változóira mutatóértékeket vesznek fel, azaz a P mutató típus a T típushoz kötődik. A mutató típus értékét általában egy memóriamenedzser-modul allokációs eljárásának meghívásakor kapja.

$ MutatóTípus = POINTER TO Típus.

A mutatóváltozók a konkrét memóriacímen kívül felvehetik a NIL értéket is. Egy NIL értékű mutató sehova sem mutat. Azonos típusú mutatókon értelmezettek az „=" és „#" ("<>") operátorok. Fontos, hogy ilyenkor a mutatók értékeit és nem az objektumokat hasonlítjuk össze.

A TopSpeed Modula-2 a mutató típusokat az Intel 80x86-os processzorcsalád és az IBM PC típusú személyi számítógépek DOS operációs rendszeréhez igazodva több bővítéssel is ellátta.

$ MutatóTípus = POINTER [Kifejezés] TO Típus.

Ha a kifejezést elhagyjuk, abszolút mutató típust deklarálunk. Egy CARDINAL típusú kifejezés megadásával bázisos mutató típust definiálunk. Az ilyen típusú mutatók csak a cím ofszet részét tartalmazzák. A cím szegmens részét a kifejezés kiértékelésével kapjuk, a mutatóval történő hivatkozáskor. Az egyedüli megkötés az, hogy a kifejezés nem tartalmazhat függvényhívást. A PC-k különböző memória modelljeihez igazodva a TopSpeed Modula-2 rendelkezik a következő standard típusokkal: NearADDRESS, FarADDRESS, illetve a következő konstansokkal: NearNIL, FarNIL. A TopSpeed Modulában a SHORTADDR standard típus 0 bázisú mutatót jelöl. Mivel eltérő szegmens—ofszet párok azonos fizikai címet jelenthetnek, ezért célszerű összehasonlítások előtt a mutatókat normalizálni.

Virtuális mutatók:

$ MutatóTípus = VIRTUAL POINTER Azonosító.

Az azonosító egy egyetlen paraméterrel rendelkező függvény, aminek a visszatérő értéke egy mutató. Az így definiált típuson csak az értékadás, paraméterként történő átadás és a hivatkozás a megengedett műveletek. A virtuális mutatókkal történő hivatkozás esetén először az „Azonosító"-val deklarált függvény hívódik meg, és a visszatérő érték szolgál a hivatkozásra.

Mutatókonstruktor:

$ Hivatkozás = „[" Kifejezés „:" Kifejezés [Név] „]".

A konstruktor két CARDINAL kifejezés kombinálásával egy szegmens és egy ofszet értékből fizikai címet állít elő. Az opcionális Név az így kapott abszolút „far" mutató típusát határoza meg. Amennyiben nincs Név, a típus alapértelmezésben FarADDRESS.

Példának okáért

A mutatók kezelésével kapcsolatban egy egész könyvet meg lehetne tölteni példákkal, ezért itt most csak a legegyszerűbb alapeseteket mutatjuk be. További példák a mágneslemez LST és MOD fájljaiban, valamint a Mesterfogások sorozatban (lesznek) találhatók.

Típusdeklarációk:

```
TYPE
  RealPtr = POINTER TO REAL;
  CardPtr = POINTER TO CARDINAL;
  RecPtr = POINTER TO RECORD
        x,y:CARDINAL;
        s:ARRAY [0..16] OF CHAR;
        rp:RealPtr;
     END;
  Link = POINTER TO NODE;
  NODE = RECORD
        key:INTEGER;
        next:Link;
     END;
 VAR
   r:RealPointer;
   list:Link;
   sr:REAL;
  NEW (list);
  ALLOCATE (r,SIZE(r^));
  r^:= 12.45;
  list^.key:= -12;
  list^.next:= NIL;
  sr:= [Seg(r):12H RealPtr]^
```

standard NEW eljárása egyébként automatikusan az ALLOCATE könyvtári eljárásra fordul le, természetesen a korrekt méretparaméterrel, ezért ha nem akarunk trükkös megoldásokat alkalmazni, célszerű ezt használni. A NEW egyetlen paramétere egy mutató változó, az ALLOCATE hívásához szükséges méretparamétert a fordító a mutató hivatkozott típusának ismeretében automatikusan szolgáltatja.

Az így létrehozott dinamikus változó értéke természetesen ilyenkor még meghatározatlan, ahhoz, hogy használni tudjuk, inicializálnunk kell, ehhez viszont ismernünk kell a dinamikus változókra hivatkozás szabályait. A dinamikus változókat ugyanúgy kezelhetjük, mint a statikus változókat, és természetesen minden olyan esetben használhatjuk őket, mint a statikus megfelelőiket. Az egyetlen különbség csupán az, hogy míg programjainkban a statikus változókra azonosítójukkal hivatkozunk, addig a dinamikus változókra a címüket tartalmazó mutatókkal. A hivatkozást ilyen esetben a mutató azonosítóját követő kalap „^" jelöli például:

szamMutato^ := 12 ;
sztringMutato^ := „dinamikus változó" ;

A már megismert hivatkozások és kvalifikált azonosítók szabadon alkalmazhatók (természetesen az adatstruktúrák és a program szintaktikájának a megsértése nélkül) például:

ModuleName.pointer^[12].ref^.record.field[i]

A dinamikus változók kezelésekor sok olyan probléma is felmerül, ami a speciális tulajdonságokból ered. Mondhatnók úgy is, nincsen öröm üröm nélkül. Nézzük mindjárt az értékadást. Ha m1 és m2 két azonos alaptípusú mutató, akkor az m2:=m1, illetve az m2^:=m1^ értékadások között lényeges különbségek vannak.

Az első esetben mindkét mutató ugyanarra az objektumra fog hivatkozni. Ez azt jelenti, hogy amennyiben m1-gyel megváltoztatjuk az objektumot, akkor ezt a változást az m2-vel való hivatkozáskor is észleljük. A kérdés csak az, hogy valóban ez volt-e a célunk. Megnyugtatásként közölhetem, hogy a kérdésre a válasz teljesen az adott alkalmazástól függ. Az ilyen típusú értékadás tehát nem hiba, de tisztában kell lennünk azzal a ténnyel, hogy egyszerre több mutató referálhat ugyanarra a memóriaterületre. Egyébként ezt a megoldást olyan esetekben használják, amikor a hivatkozott objektum a program futása során nem változik (kváziállandó vagy referenciaváltozó), és ezzel a módszerrel elkerülhetjük a felesleges duplikátumok létrehozását.

Az ilyen típusú értékadásnak a másik csapdája az, hogy — amennyiben létezett — az m2 mutató objektuma végérvényesen elveszett a program számára, és semmilyen eszközzel nem hozható vissza. Ráadásul az objektum által elfoglalt memóriaterület sem hasznosítható újra, azaz adatunkból „szemét" lett.

A második esetben az m2 által kijelölt objektumnak adunk új értéket. Ez a megoldás felel meg az idáig megszokott értékadásoknak. Gond akkor van, ha m2 még nem volt inicializálva, és ezért valahova a „vakvilágba" mutat, ami egy kis szerencsével akár a kódterület is lehet, ezzel aztán agyon is csaptuk a programunkat. Szerencsére nem C-ben, hanem Modula-2-ben programozunk, ezért nem kell kétségbeesnünk. A Modula-2 opcionális futásidejű ellenőrzése helyettünk is ügyel az ilyen jellegű hibákra, és hibajelzéssel leáll, még mielőtt nagyobb baj történne. A mutatókkal kapcsolatos műveletek esetén ugyanis a rendszer figyeli a nem inicializált, az úgynevezett NIL mutatóra történő hivatkozást. A NIL konstans az inicializálatlan vagy érvénytelen mutatóérték standard azonosítója. A NIL mutató minden mutató típussal kompatibilis.

A futásidejű memóriakezelés nemcsak új változók létrehozásából áll. Szükség van a már nem használt memóriaterületek felszabadítására is, hogy a memória-erőforrást ne használjuk el. Mint láttuk, azzal, hogy a mutatóhoz új értéket rendelünk, még nem szabadítottunk fel automatikusan memóriaterüle-

Tények, tippek, trükkök

Ne tévesszük össze az ADDRESS(a) típuskonverziót az ADR(a) standard eljárással.

Ne felejtsük el a dinamikus változókat létrehozásuk után inicializálni.

Dinamikus változók használatakor a fejlesztés során minden esetben használjuk a „run time check" opciót, így sok kellemetlen hibát fedezhetünk fel időben.

A mutato:=NIL utasítással nem memóriát szabadítunk fel, hanem szemetelünk.

A DOS korlátai ellenére is igyekezzünk profiként programozni, azaz a memóriával mint erőforrással ésszerűen gazdálkodni.

Csak az igazán időkritikus helyeken alkamazzuk a Modula-2 hardverközeli eszközeit, kerüljük a felesleges rendszerfüggőséget.

Mindig legyünk tisztában azzal, hogy az adott szituációban a mutatót vagy a hivatkozott dinamikus változót akarjuk-e kezelni.

A legtöbb Modula-2 implementációban, ha egy modulban használjuk a NEW és DISPOSE eljárásokat, akkor explicit módon importálnunk kell a Storage modul (vagy egy másik memóriamenedzser modul) ALLOCATE és DEALLOCATE eljárásait.

tet, hanem csak szemetet hagytunk magunk mögött. Itt kell megjegyezni, hogy sok rendszerben — például a Modula-3-ban — az ilyen szemetet is képes a rendszer összeszedni és újra hasznosítani. Ezt a megoldást az angol szakirodalom „garbage collection"-nek nevezi. A TopSpeed Modula-2 3.00-ás verziójában már lehetőségünk van saját köztisztasági hivatal létrehozására a programjaink számára, az ún. virtuális mutatók segítségével. A virtuális mutatók nem memóriacímet tartalmaznak, hanem egy eljárás címét, amely minden hivatkozáskor automatikusan meghívódik. A virtuális mutatók lehetővé teszik a virtuális memóriakezelés, a garbage collection, a nem darabolódó heap és még sok más igazán professzionális feladat megoldását. Alapesetben azonban nekünk kell gondoskodnunk a számunkra már feleslegessé vált memóriaterületek visszaszolgáltatásáról. Akárcsak az allokáció esetén, itt is két megoldás létezik. Az egyik a DEALLOCATE eljárás, a másik a DISPOSE standard eljárás használata. Ezeknek a használatára és alkalmazásra az ALLOCATE és NEW eljárásoknál leírtak érvényesek. A felszabadítás után a mutató NIL értékű lesz. Vigyázzunk! A mutato:=NIL megoldás azonban éppen úgy szemeteléssel jár, mint a mutató-értékadás.

Természetesen a Modula-2 nem lehetne rendszerprogramozási nyelv, ha csak a dinamikusan létrehozott változók címeit tudná kezelni. A standard ADR eljárás segítségével a tetszőleges nyelvi objektum címét megkaphatjuk. Értékadáskor pedig lehetőség van címkonstansok használatára is.

A legtöbb Modula-2 implementáció, elsősorban rendszerfejlesztési célokra, lehetőséget nyújt fix memóriacímű változók deklarálására. Az OS/2 operációs rendszer alatt futó implementációk például ezt a lehetőséget nem tartalmazzák, mivel az Intel processzorok védett üzemmódjában futó programok nem használnak fizikai címeket. Általánosságban is igaz, hogy bizonyos védett vagy védelemmel rendelkező operációs rendszerekben a hardverközeli eszközök megvalósíthatósága korlátozott. Ez abból a tényből fakad, hogy az ilyen rendszerek halálát okozná a rendszer megkerülése és a hardver direkt manipulálása. Ez persze csak konkrét rendszerekre vonatkozó korlátozás, ha kedvünk szottyan 386-os processzoron alapuló saját rendszert írnunk, alkalmazhatunk fix memóriacímű változókat.

A Modula-2 standard ADDRESS típusa minden mutató típussal kompatibilis típus, amely az adott gép fizikai címzését reprezentálja minden implementációban, így igencsak konkrét tartalommal rendelkező nyelvi absztrakció. A standard Modula-2-ben az ADDRESS típus nemcsak a mutató típusokkal, hanem a CARDINAL típussal is kompatibilis. Ez azt jelenti, hogy minden egész számokon végezhető művelet alkalmazható ADDRESS típusú operandusokra is. Így az ADDRESS típus alkalmas címaritmetika elvégzésére, aminek az eredményét tetszőleges típusú mutatóhoz rendelhetjük hozzá.

Itt meg kell állnunk egy kicsit. Mint jól tudjuk, az Intel processzorok nem lineáris, hanem szegmentált címzést alkalmaznak. A cím egy szegmens és egy ofszet részből áll, és nem éppen triviális a két 16 bites regiszter segítségével a 20 bites fizikai cím létrehozása (a szegmenscímet eltoljuk balra néggyel, majd hozzáadjuk az ofszetet). Ebből is jól látszik, hogy a PC-ken a címábrázolás nem kompatibilis az egész számok ábrázolásmódjával, ezért a PC-s Modulák ADDRESS típusa egyik egész számú típussal sem kompatibilis. Így a TopSpeed Modulában külön könyvtári eljárások szolgálnak a címaritmetika elvégzésére. Igazság szerint meg lehetett volna engedni az aritmetikai operátorok alkalmazását ADDRESS típusú operandusokon is, de úgy érzem, a deklaráltan eltérő címkezelés tisztességesebb megoldás, és jobban igazodik a nyelv filozófiájához. No nem kell megijedni, ezeknek az eljárásoknak a használata semminemű sebességcsökkenéssel nem jár, így a programunk nem veszít hatékonyságából, sőt mivel ezek a műveletek mindig normált mutatót szolgáltatnak eredményül, ezért a mutatók mint címek közötti relációs operátorok mindig helyes eredményt adnak. A szegmens—ofszet címzésmód vonatkozik minden címmegadásra.

A TopSpeed Modulában a címeket az Intel-jelölésnek megfelelően [ssss:oooo] formában kell megadni. Ezenkívül lehetőség van mutatókonstruktorok alkalmazására is. Ilyenkor két CARDINAL típusú kifejezés segítségével „far" mutatót hozhatunk létre, aminek típust is adhatunk egy típusazonosító révén. Természetesen a TopSpeed fordító támogatja a különböző memóriamodellek használatát is. Erre a célra (a minket most érintő mutató témakörben) a nyelvet kibővítették a NearAddress, FarADDRESS, illetve NearNIL és FarNIL standard azonosítókkal. A JPInál — úgy látszik — nagyon is tisztában vannak a mutatók fontosságával, mivel a TopSpeed Modula-2 még egy mutató típussal, az úgynevezett bázisos mutatóval rendelkezik. Egy mutató definíciójában megadhatunk egy CARDINAL típusú kifejezést, és ezzel a mutató csak az ofszet részt tartalmazza, a szegmens rész a kifejezés kiértékelésével kapható meg. Az ilyen bázisos mutatók a relokálható objektumok kezelésében játszanak szerepet. És van még egy nyelvi bővítés a mutatóknál, a SHORTADDR standard típus, ami minden bázisos mutatóval kompatibilis. Ezenfelül a TopSpeed könyvtár a standard „Storage" modulon kívül más memóriamenedzsereket is tartalmaz a különféle heapek kezelésére.

A Modula-2 nyelvi elemeit bemutató sorozatnak ez a része nem véletlenül sikeredett ilyen hosszúra. A dinamikus adatstruktúráknak és a mutatóknak a programozás mesterfogásait bemutató részekben alapvető szerepük lesz. Az olyan absztrakt adatstruktúrákat, mint a listák, a sorok, a fák, a dinamikus tömbök és az általános halmazok mind mutatók segítségével valósítjuk majd meg. De említhetném még az adatbázis-kezelő és az alkalmazási környezet megvalósítását is. Addig is bemelegítésként a lemezmellékleten az LST és MOD fájlokban néhány egyszerűbb példa található a mutatók kezelésére.

Villányi László

Pascal és C

A Modula-2 mutató típusok a Pascal kiterjesztett változatai. A Pascalban nincs lehetőség a rendszerprogramozásra a fizikai címek manipulálásával. Nincsen ADDRESS típus, címaritmetika, ADR standard eljárás, és a dinamikus memóriakezelő eljárások is egyszerűbbek. A standard Pascalban csak „new" eljárás volt, „dispose" nem. Nem is beszélve a Modula-2 külső könyvtári moduljai segítségével megvalósított memóriakezelésről. A C és a Modula-2 lehetőségei a dinamikus változók kezelésében megegyeznek. A C kevesebb, hardverprogramozáshoz szükséges lehetőséget tartalmaz, de ez az esetek nagy részében nem jelent problémát. A C, mint általában, a mutatók esetében sem kényes a kompatibilitási szabályok betartására.

Clipper-tippek

Ecsetvonások balra-jobbra

A múlt havi részben addig eljutottunk, hogy körülbelül mire használhatjuk a TBROWSE objektumokat, és mik azok a lehetőségek, amelyekkel élhetünk adatmegjelenítés közben.

TBROWSE objektumot kétféleképpen tudunk létrehozni. Egyik lehetőség a TBrowseNew( ) függvény használata, másik pedig a TBrowseDB( ) függvényé. Mindkét függvénynek négy paramétert adhatunk meg, mégpedig az adatmegjelenítő táblázat négy sarokpontját (sorrend: felső sor, bal oszlop, alsó sor, jobb oszlop). A két megoldás között alapvető különbség nincsen, ami mégis van, arra még visszatérünk.

A fentebb említett két függvény visszatérési értéke egy objektum típusú változóban tárolható. Ha például mybrow egy objektum típusú változó, akkor érvényes a következő értékadás: mybrow = TBrowse New (p1, p2, p3, p4). Ezután a mybrow egy TBROWSE osztályú objektumot tartalmaz, amelynek változóira így hivatkozhatunk mybrow:változónév, eljárásait pedig így hívhatjuk meg: mybrow:eljárásnév(paraméterek).

A két létrehozó függvény csak létrehozza az objektumot, de hogy az valamit meg is jelenítsen, arról már nekünk kell gondoskodnunk. Ez a gondoskodás azt jelenti, hogy oszlopokat kell definiálnunk a táblázat számára. Az egyes oszlopokat leíró struktúrákat TBCO LUMN osztályú objektumokban kell elhelyezni, és ezeket az objektumokat lehet aztán a TBROWSE objektumhoz hozzáadogatni. Egy TBCOLUMN objektumot az addColumn( ) eljárással tudunk a TBROWSE objektumhoz hozzáadni. Az eljárásnak paraméterként kell átadni a TBROWSE objektumra mutató változót. A TBCOLUMN osztályú objektumoknak mindösszesen 10 külső változójuk van. Az oszlop definiálása abból áll, hogy létre kell hozni egy új objektumot, majd ennek a 10 változónak értéket kell adni. Új TBCOLUMN osztályú objektumot a TBcolumnNew( ) függvénnyel lehet létrehozni. A függvénynek két paramétert adhatunk át, mégpedig a heading és a block változó kezdeti értékeit. Az összes változó leírását tartalmazza az 1. táblázat.

Ha egy TBROWSE objektumot létrehoztunk, akkor a változóinak beállításával tudjuk befolyásolni a kijelzés egyes jellemzőit. A létrehozó függvénynek paraméterként adtuk meg a kijelzés sarokkoordinátáit, de ezeket, ha a szükség úgy hozza, akkor megváltoztathatjuk. Az nBottom nevű változó az ablak alsó, az nTop nevű pedig az ablak felső sorát tartalmazza. A bal oldali oszlopkoordináta a nLeft, a jobb oldali az nRight változóban van tárolva. Ha ezeket meg akarjuk változtatni, akkor nincs más dolgunk, mint a megfelelő változót átírni a megfelelő értékre. A változók átírása után meg kell hívni a refreshAll( ) nevű eljárást, amely a beállított paramétereknek megfelelően újrarajzolja a teljes TBROWSE ablakot.

A kijelzés során a TBROWSE objektumnak saját logikai kurzora van. Ez a kurzor nem egy képernyőpozícióra, hanem az objektum egy mezőjére mutat. Az aktuális kurzorpozíció sorát a rowPos, az oszlopát pedig a colPos változó mutatja. Ezekben a változókban nem képernyőkoordináták vannak, hanem oszlop- és sorkoordináták. Ha a colPos változó értéke 2, a rowPos változó értéke pedig 3, akkor az objektum kurzora a második oszlop harmadik sorában van. Az a mező, amelyiken az objektum kurzora van (aktuális mező), automatikusan alászíneződik. Ha viszont az autoLite nevű logikai változóba .F. értéket írunk, akkor az alászínezés automatikussága megszűnik. Ebben az esetben a hilite( ) eljárással tudjuk alászíneztetni az aktuális mezőt, és a dehilite( ) eljárással tudjuk levenni róla az alászínezést. Amikor egy új objektumot létrehozunk, akkor az autolite változó értéke mindig .T., egészen addig, amíg át nem írjuk.

Az TBROWSE objektum kurzorát exportált eljárások segítségével nekünk kell pozicionálnunk. Ez úgy történik, hogy a táblázat kirajzolása után elkezdjük figyelni a billentyűzetet, és ha mondjuk az UP(8) gombot megnyomják, akkor meghívjuk az up( ) nevű eljárást, ami az objektum kurzorát egy sorral feljebb viszi. Az 3. táblázatban ismertetett kurzormozgató eljárások fel vannak készítve arra, hogy szükség esetén scrollozzák a táblázatot bármely irányban. Mivel a táblázat kirajzolása után a billentyűzet figyelése a mi dolgunk, megtehetjük azt is, hogy bizonyos gomboknak különleges jelentőséget programozunk. Legegyszerűbb esetben a kurzormozgató billentyűkön kívül nem is kell mást figyelni. Komplikáltabb esetben egymásba ágyazott TBROWSE megjelenítéseket vezérelhetünk és/vagy mindenféle látványos egyéb tevékenységet hajthatunk végre.

Amikor a vertikális irányú kurzormozgató eljárások elérik vagy túllépik a kijelzett rész határait, akkor a kijelzés aktualizálásáról is gondoskodni kell. Ennek a szervezését az objektum belső függvényei megoldják. Ha a legalsó kijelzett sorban van a kurzor, és meghívjuk a down( ) eljárást, akkor az egész táblázat egy sornyival feljebb gördül. A legfelső sor eltűnik, alul pedig megjelenik az új sor. A kijelzés aktualizálásáról gondoskodik ugyan az objektum, de hogy ezt hogyan tegye, azt már nekünk kell megadnunk. E célra három külső változót tartalmaz egy TBROWSE objektum. A skipBlock nevű változóba egy kódblokkot kell elhelyeznünk. Ez a kódblokk lesz aktivizálva mindig, ha a kijelzett táblázatot scrollozni kell. Argumentumként kapja a kódblokk, hogy hány sornyit kell görgetni a táblázaton. Negatív argumentum lefelé, pozitív pedig felfelé görgetést ír elő. Ha az előző példa szerint a táblázat alsó sorában van a kurzor, és meghívjuk a down( ) eljárást, akkor az objektum aktivizálja a skipBloc nevű változójában lévő kódblokkot, argumentumként +1 értéket ad át neki. A kódblokknak nem kell aktualizálnia a kijelzést, csak a kijelzett táblázat sormutatóját (rekordmutatóját) kell mozgatnia (adatbázis megjelenítése esetén egyszerűen: SKIP). A másik két ilyen változó a goBottomBlock, amelyet a goBottom nevű kurzormozgató eljárás és a goTopBlock, amelyet a goTop nevű kurzormozgató eljárás használ. Itt kell kitérni a TBrowseNew( ) és a TBrowseDB( ) létrehozó függvények közti különbségre. A TBrowseDB( ) függvénnyel létrehozott objektumoknál az imént említett három változóba automatikusan beíródnak a következő alapértelmezett értékek: go-

TopBlock := {||GO TOP}; goBottomBlock := {||GO BOTTOM}; skipBlock := {|x|SKIP x}.

A colorSpec nevű változóban egy színtáblázatot adhatunk meg. Ha egy TBROWSE objektumot létrehozunk, akkor ide a SETCOLOR( ) függvény visszatérési értéke kerül, de ezt bármikor megváltoztathatjuk. A táblázatot karakterstring formában kell megadni, ugyanúgy, mint a SETCOLOR( ) függvény esetében. Az elsőként megadott színben jelennek meg a táblázat mezői. A másodikként megadott színt veszi fel az aktuális TBROWSE kurzorpozíció által jelölt mező. A változóba írt többi színpárost a TBcolumn objektum colorBlock nevű változója jelölheti ki. Mindig, mielőtt egy új érték kijelzésre kerül, végrehajtódik a megfelelő oszlopobjektum colorBlock nevű változójában lévő kódblokk. A kódblokk visszatérési értéke egy kételemű numerikus tömb. A tömb első eleme annak a színnek a sorszáma a colorSpec táblázatból, amely az adott változó színe lesz. A tömb második eleme által jelölt színű lesz az adott mező, ha az objektum kurzora rákerül. A colorBlock nevű változóban lévő kódblokknak végrehajtáskor átadódik a mező értéke. Így lehetőség van például negatív számok vagy más jellegzetes adatok eltérő színben való megjelenítésére.

Ha a right( ) kurzormozgató eljárást olyankor hívjuk meg, amikor az objektum kurzora a jobb oldali szélső oszlopban van, akkor a kijelzett oszlopok egy lépéssel balra gördülnek. A bal szélen lévő oszlop eltűnik, helyére az addig másodikként szereplő oszlop kerül, jobb oldalt pedig megjelenik egy új oszlop. A freeze nevű változó használatával lehetőség nyílik a bal oldali oszlop befagyasztására. Ekkor balra scrollozás esetén a befagyasztott oszlopok nem gördülnek ki a képernyőről. A freeze változóba numerikus értéket írhatunk. Balról számolva a beírt számú oszlopot befagyasztja.

Az egyes oszlopokat elválasztó karaktereket a TBROWSE objektum colSep változójába kell beírni. A TBCOLUMN objektumok azonos nevű változója felülbírálja ezt a beállítást. A táblázatot a lábrésztől elválasztó karaktert az oszlopobjektum footSep változójába kell írni, a fejléctől elválasztó karaktert pedig a headSep változóba.

A TBROWSE objektumoknak van egy colorRect( ) nevű eljárásuk. Ennek segítségével átszínezhetjük a táblázat egy négyzet alakú részletét. Az eljárásnak két tömböt kell átadni paraméterként.

Fridl György

1. táblázat. TBROWSE objektumok változói

Név	Típus	Leírás
nBottom	Num.	Az ablak alsó sorkoordinátája
nTop	Num.	Az ablak felső sorkoordinátája
nLeft	Num.	Az ablak bal oszlopkoordinátája
nRight	Num.	Az ablak jobb oszlopkoordinátája
rowCount	Num.	Az ablak sorainak száma (csak olvasható)
colCount	Num.	Az ablak oszlopainak száma (csak olvasható)
rowPos	Num.	Az aktuális mező sorának száma
colPos	Num.	Az aktuális mező oszlopának száma
colSep	Kar.	Az egyes oszlopokat elválasztó karakter
headSep	Kar.	A táblázatot a fejléctől elválasztó karakter
skipBlock	Kódbl.	A kódblokk, amely újrapozicionálja az adatállomány rekordmutatóját
goTopBlock	Kódbl.	A kódblokk, amely az adatállomány elejére pozicionálja a rekordmutatót
goBottomBlock	Kódbl.	A kódblokk, amely az adatállomány végére pozicionálja a rekordmutatót
hitTop	Log.	Igaz, ha a rekordmutató az adatállomány első rekordjára mutat
hitBottom	Log.	Igaz, ha a rekordmutató az adatállomány utolsó utáni rekordjára mutat
autoLite	Log.	Hamis értéke tiltja az aktuális mező automatikus alászínezését
colorSpec	Kar.	Színsorozat a megjelenítéshez. Leírása a szövegben.
freeze	Num.	A bal oldalt nem scrollozható oszlopok száma
stable	Log.	.F. a megjelenítés aktualizálása közben, egyébként .T.
cargo		* Szabadon használható változó

2. táblázat. TBROWSE objektumok kurzormozgató eljárásai

left()	Az objektum kurzorát balra mozgatja egy oszloppal
right()	Az objektum kurzorát jobbra mozgatja egy oszloppal
up()	Az objektum kurzorát felfelé mozgatja egy sorral, ha kell, újrarajzolja a táblázatot
down()	Az objektum kurzorát lefelé mozgatja egy sorral, ha kell, újrarajzolja a táblázatot
end()	A jobb oldali legszélső kijelzett oszlopba teszi a kurzort (nem scrolloz)
home()	A bal oldali legszélső kijelzett oszlopba teszi a kurzort (nem scrolloz)
pageDown()	rowCount sornyival előre mozdítja a rekordmutatót, és újrarajzolja a táblázatot
pageUp()	rowCount sornyival hátra mozdítja a rekordmutatót, és újrarajzolja a táblázatot
goBottom()	Végrehajtatja a goBottomBlockban lévő kódblokkot, és újrarajzolja a táblázatot
goTop()	Végrehajtatja a goTopBlockban lévő kódblokkot, és újrarajzolja a táblázatot
panEnd()	A jobb szélső oszlopba teszi a kurzort
panHome()	A bal szélső oszlopba teszi a kurzort
panRight()	Egy oszlopnyival jobbra scrollozza a kijelzést
panLeft()	Egy oszlopnyival balra scrollozza a kijelzést

3. táblázat. TBROWSE objektumok exportált eljárásai

addColumn(oszlopobjektum)	TBCOLUMN objektum (oszlop) hozzáadása a TBROWSE objektumhoz
setColumn(sorszám,oszlopobjektum)	Oszlopobjektum megváltoztatása (cseréje)
getColumn(sorszám)	Oszlopobjektum lekérdezése
hilite()	Az aktuális mező alászínezése
dehilite()	Az aktuális mező alászínezésének levétele
refreshCurrent()	Az aktuális mezőt tartalmazó sor újrarajzolása
refreshAll()	A teljes táblázat újrarajzolása
configure()	Az objektum újrakonfigurálása
colorRect(sarokponttömb,színtömb)	Négyzet alakú terület átszínezése

4. táblázat. TBCOLUMN objektumok változói

block	Kódbl.	A mező tartalmát visszaadó kódblokk
colorBlock	Kódbl.	A mező színének sorszámait kételemű numerikus tömbben visszaadó kódblokk
defColor	Num. tömb	A normál mező színeinek sorszámait tartalmazó kételemű numerikus tömb
colSep	Kar.	Az oszlopokat elválasztó karakter (felülbírálja a TBROWSE objektum colSep változóját)
footSep	Kar.	A lábrészt a soroktól elválasztó karakter
headSep	Kar.	A fejrészt a soroktól elválasztó karakter
footing	Kar.	A lábrész szövege
heading	Kar.	A fejrész szövege
width	Kar.	Az oszlop szélessége

Cluster Buster

Szektorsemlegesnek nem mondható

Magyarországon új (és az eddigi legnagyobb) számítógépes vírusfertőzési hullám söpör végig. Főszereplője a Cluster Buster vírus. (Egyéb ismert elnevezései: DIR2/FAT, D2, DD.) Gyors terjedésének fő oka, hogy a változásokat figyelő semmilyen vírusmegelőző rendszer nem veszi észre. A vírusspecifikus kereső és ölő programoknak is csak a legfrissebb verziói ismerik.

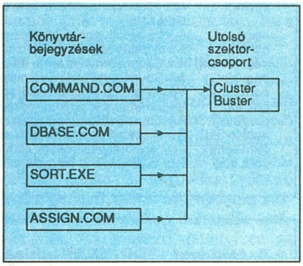
Nem árt felidézni, hogy a szakirodalom a vírusokat működési módjuk fő jellemzője alapján két nagy csoportba sorolja:

1. Bootvírusok. A rendszer betöltési (bootolási) folyamatában aktivizálódó vírusok a merevlemez partíciós tábláját vagy bootszektorát fertőzik meg, a floppylemezeken pedig a bootszektorba íródnak (ilyen pl. a Stoned, a Michelangelo stb.).

2. Fájlvírusok. A futtatható állományok indításakor aktivizálódó vírusok az állományokat változtatják meg, egészítik ki a vírus programjával. A fertőzött programok elindításakor először a vírus fut le, majd ezt követi az eredeti program (ilyen pl. az 1701, a Yankee Doodle stb.).

Ritkábban ugyan, de az említett két csoport kombinációja is előfordul (pl. Invader).

A most megjelent Cluster Buster azonban egyik csoportba sem sorolható. Terjedése új elven alapszik, amelynek lényege, hogy a vírus a futtatható (COM és EXE) állományok könyvtárbejegyzését változtatja meg, úgy hogy az operációs rendszer először a vírus testét tartalmazó szektorcsoportot (clustert, allokációs egységet) töltse be, és azt futtassa le. Ezen kódrészlet működtetése után természetesen az eredeti program is végrehajtódik. Ebből az következik, hogy elegendő a vírus testét tartalmazó szektorcsoportnak a DOS partíciójú merevlemezen egyetlen helyen jelen lennie. (Olyan lemezeknél, ahol a „szektorcsoport" csak egyetlen szektort tartalmaz, a vírus két szektort foglal el.) A Cluster Buster az utolsó szektorcsoportban található. A fertőzött futtatható állományok könyvtárbejegyzései oda mutatnak (lásd az ábrát).



A vírus tehát a könyvtárbejegyzésekben az első szektorcsoportra mutató számot módosítja a felülírt szektorcsoport javára. Az eredeti első szektorcsoport számát a könyvtárbejegyzésen belül egy nem használt területre írja. A vírus írója gondolt a vírusirtókra is. Az eredeti első szektorcsoport-számot ugyanis kódolva, titkosítva írja vissza a könyvtárbejegyzés nem használt területére. A titkosítás függ egyrészt attól, hogy az állomány könyvtárbejegyzése hányadik az aktuális alkönyvtárban, másrészt pedig függ az utolsó szektorcsoport két bájtjától.

A vírus memóriában maradásának (rezidenssé válásának) megoldása is újdonságszámba megy. A vírus úgy épül be ugyanis a memóriába, hogy egyetlen megszakító (interrupt) vektort sem állít magára. Ezzel szemben beépül az eszközmeghajtók (device driverek) láncolatába, így a megszakítók változását, működését figyelő rezidens vírusfigyelő programok nem képesek észrevenni. A vírus rezidensen a memóriában lehet, mégis minden állomány eredeti, fertőzetlen állapotában mutatkozik.

A vírus működésének vannak bizonyos korlátai:

— Semmilyen hálózaton nem képes terjedni és ott állományokat megfertőzni.

— Kizárólag az operációs rendszer által kezelt meghajtókon tud fertőzni. Nem terjed például a Disk Manager, SpeedStore programokkal formázott lemezek azon partícióin, amelyeket a DOS egyébként nem lát (D:, E:, ...).

— A vírus hibásan működik abban az esetben, ha egy szektor nem 512 bájtot tartalmaz, vagy ha a partíciónak 65535-nél több szektora van.

A vírus eltérő tüneteket, jelenségeket idéz elő, attól függően, hogy aktív-e (rezidens-e a memóriában).

a) Ha a vírus nincs a memóriában, nem rezidens, akkor:

— A fertőzött állományok futtatásával a vírus bejut a memóriába, majd az eredeti program hajtódik végre.

— A fertőzött programok másolásával nem az eredeti állományt, hanem magát a vírust másoljuk. Az állomány hossza sem az eredeti, hanem egy szektorcsoport méretének felel meg: 1024, 2048, 4096, ...

— A lemez szektorcsoportjait, FAT-bejegyzéseit ellenőrző, módosító programok (ChkDsk, NDD. ...) FAT-hibára panaszkodnak.

b) Ha a vírus már a memóriában található, akkor semmilyen gyanús jelenséget nem okoz.

A vírus a lemez utolsó (vagy utolsó két) szektorcsoportját felülírja. Ezt minden ellenőrzés nélkül teszi, s ami ott volt, az elvész. Ezen túlmenően más romboló tevékenysége nincs, viszont a lemez FAT-bejegyzéseit, a szektorcsoportok elhelyezkedését módosító bármilyen program használata helyreállíthatatlan károkat okozhat.

A vírust a ChkVir program 5.22 feletti változatai felismerik és irtják, ugyanígy a McAfee-féle Scan/Clean programcsomag is a 84-es verziótól kezdve.

Leitold Ferenc

A Magyar Vírusirtó Csoport (Hungarian VirusBuster Team) ingyenes tanácsadó szolgálata az alábbi telefonszámokon érhető el: 166-4011/2752-es mellék vagy 133-1670.

Rejtvényverseny

Számnévszemantika mint agytorna

Hajlamosak vagyunk rá, hogy amit megszoktunk, azt tartsuk természetesnek, univerzálisnak, általánosan emberinek. Mostani feladatunk is arra figyelmeztet, hogy nem előny, ha valakinek a gondolkozását túlságosan behatárolják korábbi beidegződései, ha képtelen szabadulni saját, egyedül lehetségesnek képzelt gondolati formáitól.

Egy ismeretlen nyelven a következő összetett számnévvel találkoztunk: „jiuqian ling wu yü sanqian ling sishi wan". Feladatunk egyrészt annak kiderítése, hogy ez a számnév mit jelent, másrészt ezen keresztül megérteni — természetesen további információk bevonásával — a számnevek felépítésének egész rendszerét ezen az ismeretlen nyelven. A megfejtéshez a könnyebbség kedvéért két jelölésbeli módosítást vezetünk be:

(1) Az egyszavas összetett számneveken belül megjelöljük az elemi összetevők határvonalát. Ezt azzal érhetjük el, hogy minden elemi összetevőt nagybetűvel kezdünk: JiuQian, SanQian, SiShi.

(2) A többszavas összetett számneveket aláhúzásjellel kapcsoljuk egybe. Ahhoz, hogy ennek értelmét belássuk, pillantsunk lejjebb: a kiegészítő információkat aritmetikai egyenlőségek formájában adjuk meg, ennek áttekinthetőségéhez pedig világosan látnunk kell, hogy mettől meddig tekintendő valami egyetlen argumentumnak.

Arra kérjük az olvasót, hogy megoldás közben készítsen olyan feljegyzéseket is, amelyekből utólag kiderül, milyen sorrendben használta fel a rendelkezésére álló információkat. E felkonferálás után kezdődjék a feladat.

Első információcsokor

Néhány számnévről első ismerkedésünk alkalmával kiderült, hogy szorzatukat már a kisiskolások is ismerik az „egyszeregyből", hiszen a Ba, Er, Qi, Wu számnevek 10-nél nem nagyobb számokat jelölnek. Írjuk fel számjegyek helyett számnevekkel ezeket a szorzatokat:

Wu * Er = Shi (1)
Ba * Wu = SiShi (2)
Er * Qi = ShiSi (3)

Második információcsokor

További vizsgálódásaink során fény derült rá, hogy bizonyos aritmetikai összefüggéseket hogyan lehet kifejezni ennek a nyelvnek a számneveivel.

Ba + Ba = ShiLiu (4)
Wan * Wan = Yü (5)
Yi * Yi + Yi = Er (6)
Qi * Qi + Yi = WuShi (7)
Wu * Wu * Ba = ErBai (8)
Yü — Bai_Wan = JiuQian_JiuBai_Wan (9)
YiBai * YiBai = Wan (10)
ShiYi + ShiYi = ErShi_Er (11)
Bai_Wan — Wan = JiuShi_Jiu_Wan (12)
JiuShi + JiuShi = YiBai_BaShi (13)
ShiLiu * ShiLiu = ErBai_WuShi_Liu (14)
YiQian * YiQian = Bai_Wan (15)
Jiu * Jiu + Jiu = JiuShi (16)
Qi * Jiu + ErShi_Qi = JiuShi (17)
Qi * ShiYi * ShiSan = YiQian_Ling_Yi (18)
ShiYi * ShiYi * ShiYi = YiQian_SanBai_SanShi_Yi (19)
ShiLiu * ShiLiu * ShiLiu = SiQian_Ling_JiuShi_Liu (20)

A jelölésben a szorzás és az összeadás műveleti jelei között a szokásos prioritási konvenciókat alkalmaztuk, amely szerint a szorzásjel „erősebben köt", mint az összeadásjel.

Kérdések:

1. Mit jelent ezen a nyelven a következő összetett számnév: JiuQian_Ling_Wu_Yü_SanQian_Ling_SiShi_Wan?

2. Mely egyenletek jelentették önnek a legtöbb információt a megoldás során? Miért?

3. Hogy lehetne segédjelek (zárójelek és műveleti jelek) bevezetésével „kijavítani" a természetes nyelv jelölésrendszerét, hogy explicit módon egyértelművé tegyük az implicit módon beléjük értett műveleteket? Hogyan kellene matematikailag precízen felírni az egyszavas (például SiShi, ShiSi) és többszavas (például JiuShi_JiuBai_Wan) számnevek természetes nyelvi alakjait?

4. Hogyan lehet kifejezni ezen a nyelven a 142857 számot, továbbá e számnak azokat az egész számú többszöröseit, amelyek 1 milliónál kisebbek? (Vigyázat! Érdemes még egyszer megnézni a nagy számok szerkezetét!)

Vajon milyen nyelvről lehet szó?

Extra feladatok extra jutalomért:

+1. Milyen gondolatmenet szerint működne az a program, amely értelmezni tudná és számokkal le tudná írni ezeket az összetett számneveket?

+2. Hogy működne (vázlatosan) az a program, amely számokból generálni tudna hasonló számneveket ezen az ismeretlen nyelven?

A megfejtéseket a szerkesztőség címére (1251 Budapest, Pf. 71) kérjük beküldeni 1992. január 31-ig.

Az értékelésben létraversenyszerűen összesítjük a szerzett pontokat. A legjobb megoldásokat közölni szeretnénk az Alaplapban, illetve a mágneslemez mellékleten.

Az éllovasokat félévenként az Alaplap, illetve a Floppy.Lap éves előfizetésével díjazzuk. Az extra feladatokért külön díjazás jár: 3-3 szabadon választott shareware lemez!!

A létraverseny állása 1991. december 10-én változatlanul: CsurgayPéter 30, Dezső András 30, Geszat Zoltán 30, Süle Gábor 40, Vágó Dániel—Láng Attila D. 30.

Vargha Dénes

Ki teszi rá a vírust?

Lemezrejtelmek

Novemberi számunk ráadásként adott lemezmelléklete, a játék különszám meglepően sok reklamációt okozott. A kísérőlevéllel visszaküldött lemezek némelyike kifejezetten talányos volt számunkra, a hiba okának felderítése nem is minden esetben sikerült. A tanulságok azonban mindenkit érdekelhetnek.

Kezdjük az egyszerűbb esetekkel. Idézet egy zalaegerszegi levélből: *„Mellékelten visszaküldöm az Alaplap 1991. novemberi számának Játék különszám című mellékletét, azzal, hogy a számítógép olvasási hibát jelzett, nem tudott rámenni az a: meghajtóra."* Az e levélhez csatolt lemez teljesen hibátlan volt, s a téves reklamáció valószínűleg a felhasználó járatlanságára vagy figyelmetlenségére vezethető vissza. Minden bizonnyal nem olvasta el az **!OLVASS.EL!** szövegállományt, hanem egyenesen belépett az ARKEGA alkönyvtárba, s elindította az ARKEGA#.EXE önkicsomagoló állományt, amely elkezdte ugyan a tömörített fájl kicsomagolását, de hamarosan elakadt, hiszen a floppyn már nem volt annyi hely, hogy az Arkega játék kicsomagolva ráférjen. (Ha lett volna, mi is úgy tettük volna fel, de helyhiányból ezt az egyet tömörítettük.) A gép ezért teljesen logikusan **íráshibát** jelzett (write error), nem pedig olvasási hibát, ahogy azt a feladó nekünk tévesen írta. Az összes többi játék futtatható volt a floppyról közvetlenül is, csak az Arkegát kellett rátenni előbb egy elegendő szabad hellyel rendelkező winchesterre vagy floppyra, és ott kibontani.

A fenti eset jó példa arra, hogy mielőtt bármilyen programot elindítanánk, érdemes egy pillantást vetni (view) a mellette lévő OLVASS.EL (readme) állományra. Sok felesleges munkától és bosszankodástól megkímélhetjük magunkat.

Egyeseket talán megzavart az, hogy miközben a szokásos lemezmelléklet „automata" működésű, a játékmellékleten nem volt szükséges a „keretjátékot" is elhelyeznünk a külön önálló életet élő, egymáshoz nem tartozó programokhoz. Másik feltételezésünk, hogy a játéklemez esetleg olyanok figyelmét is felkeltette, akik eddig a másik lemezmellékletet csak nyersanyagként kezelték, vagy eleve továbbadták, így a kezdő géphasználók tipikus tévesztései most lökésszerűen felszínre kerültek.

Nem ilyen egyszerű viszont azoknak a lemezeknek az esete, amelyeket úgy kaptunk vissza, hogy nem volt rajtuk semmi. Szó szerint semmi, még **formázás**, tehát szektorbeosztás sem — legalábbis PC-norma szerinti nem —, s ez különösen azért talányos, mert — egy kis üzemi titkot kifecsegve — a játéklemezeket kétszer is formáztuk, mivelhogy a korábbi számok eladatlan (remittenda) példányaiból kiszedett lemezek felülírásával sikerült ezt a kis ráadást összehozni. Rejtélyes tehát, hogyan kerülhetett például a Veszprémi Tervező Vállalathoz egy „szűzzé visszaváltozott" lemez. Csak tippelni tudunk, hogy a lemezt valahol esetleg erős elektromágneses hatás érte, nem PC-formátumban újraformázták, vagy általunk teljesen ismeretlen módszerrel „leradírozták". Ha valaki hiteles magyarázattal tud szolgálni, azt mindenki okulására szívesen leközöljük.

Egyes visszahozott lemezek mechanikai sérüléseinek eredetére sem tudtunk fényt deríteni. Függetlenül viszont a hibát előidéző októl, a visszaküldött lemezek fejében mi természetesen minden reklamálónak megküldjük, vagy ha személyesen jön, azonnal kicseréljük a lemezmellékletet.

Más a helyzet a vírusosan visszaküldött lemezekkel. Minden hónapban akad néhány olvasónk, aki lemezmellékletünkön vírust talált. Nem azt vonjuk kétségbe, hogy valahogy vírus került a lemezre, mert az tény, ha egyszer ott van. Csak hát az a vírus nem a gyártás, hanem a használat során mászott oda! Néhányan vissza is küldik ezeket a lemezmellékleteket. Az egyik vidéki könyvtár egész évfolyamnyi fertőzött lemezt juttatott el így hozzánk, előfizetésének lemondása kíséretében. Vagy idézünk például egy friss és konkrét panaszt: „Sajnálattal tudatom, hogy a 11. számukban lévő lemezek vírusosak! A vírust a Scan 82 találta a bootszektorban. A vírus neve Michaelangelo [Mich]. Részvéttel egy olvasó."

Kedves vírusgazdák! Bár Kis János ismert vírusvadász kollégánk közvetlen közelünkben kifejtett ismeretterjesztő tevékenysége révén némi vírusismerettel mindannyian fel vagyunk vértezve, nem tartjuk magunkat a vírusokkal szemben sebezhetetleneknek. Velünk is előfordulhatott volna, hogy figyelmetlenségből, viszonylag új vírus felbukkanásakor, vagy „moral insanity" következtében a lemezmelléklet mintapéldánya vírussal fertőződik. Ebben az esetben viszont nem egy-két lemezmelléklet lett volna vírushordozó, hanem mind a 11 000 példány. Ennek egyszerű technológiai oka van: a mintalemezről a sokszorosítás fizikai másolással történik. A másolatokra mindig csak az kerülhet rá, ami a mintapéldányon is rajta van — nem több és nem kevesebb. A sokszorosított lemez pedig számítógéppel már csak a lehegesztett átlátszó tasak felbontása után, az olvasónál találkozhat, ott fertőződhet meg, tapasztalataink szerint leggyakrabban valamelyik bootvírussal. Ezeknél ugyanis akár el sem kell indítanunk a lemezről egy programot, elég, ha katalógust kérünk róla (dir), és a bootvírus a gépről már rá is telepedett a floppyra. Néhány géphasználó pedig sokáig észre sem veszi, ha teli van új bootvírusokkal!

Másfél év alatt az Alaplap lemezmellékletéből egyetlen példány sem került ki vírussal fertőzötten. Ám az ördög — és az emberi tisztességtelenség — nem alszik, ezért a jövőben a lemezt sokszorosítás előtt nagyon friss vírusismerettel rendelkező szakértővel külön is ellenőriztetni fogjuk. A régi vírusokkal magunk is megbirkóznánk, de rendszeresen jönnek az újak — ahogy jött legutóbb a Michelangelo vagy a D2 —, és felderítésük annál nehezebb, minél agyafúrtabb új programozási technikát találnak ki „nemes lelkű" szerzőik.

Faklen Pál

Számítógépek sakkversenye

Mi történt Vancouverben?

Az Alaplap novemberi számának sakk témájú cikkeihez kapcsolódva néhány érdekesség jutott eszembe a Vancouverben rendezett mikroszámítógépes sakkvilágbajnokságon tapasztaltakról. Négy amerikai, három angol, két-két holland, német, francia és egy orosz programozó indította programját, a magyar színeket pedig az én Brainstorm nevű programom képviselte.

A vancouveri résztvevők és a végső eredménytáblázat:

1.	Gideon (Hollandia)	6
2.	M Chess (USA)	5,5
3–4.	Mephisto (Anglia)	5
3–4.	The King (Hollandia)	5
5.	Spracklen X (USA)	4,5
6–7.	Hiarcs (Anglia)	4 (*)
6–7.	Now (USA)	4 (*)
8–10.	Brainstorm (Magyarország)	3,5
8–10.	Nightmare (Németország)	3,5 (*)
8–10.	Echec (Franciaország)	3,5
11.	Patzer (Németország)	3 (*)
12–14.	Cumulus (Franciaország)	2,5
12–14.	Woodpusher (Anglia)	2,5 (*)
12–14.	Innovation (USA)	2,5 (*)
15.	Centaur (Szovjetunió)	1 (*)

(*) Plusz 1 pont, amit akkor kap a versenyző, ha nincs játszmája.

Lenyűgöző volt a hardver, amelyen a programok futottak. Volt 31 MHz-es RISC-processzoros gép, 60 MHz-es 68030-as gép, 33 MHz-es 486-os, SPARC Workstation stb. A PC-s programok mind 486-oson futottak, kivéve az enyémet. Ezt ugyanis egy speciális 68020-as kártyára írtam, amelyet egy 6 MHz-nél nem gyorsabb AT-be bedugva lehet működtetni. A rendezőknek nem kis utánajárásba került, hogy ilyen lassú gépet szerezzenek. Furcsa módon ennek a bérleti díja volt a legmagasabb a gépek között. (Nyilván azért, mert ez ott már ritkaságnak számít.)

Furcsa volt, hogy egy programozóteam megduplázódjon, és két programmal jelenjen meg, ami a francia Echec és Cumulus esetében történt. A két program külső megjelenésében semmi különbség nem volt. Ez állt egyébként a holland programokra is: Gideon és The King sem volt megkülönböztethető ránézésre. A verseny során nagy jelentőségűnek bizonyult, hogy csak fél év telt el az előző verseny óta. Ennek volt ugyanis köszönhető, hogy trónfosztották a Mephistót, amely már hét éve egyeduralkodó volt a mikroszámítógépes sakkvilágbajnokságokon. Az történt ugyanis, hogy ennyi idő alatt nem tudtak új programmal előrukkolni, és a lyoni verzióval indultak. Ezt viszont már az előző verseny óta a kereskedelemben is forgalmazzák, így az ellenfelek hozzájuthattak, tesztelhették.

Ezt tette a holland Jan Louwman is, aki a győztes Gideon programot kezelte. Mintegy 1000(!) órán át tanulmányozta a Mephisto Lyont, és olyan ördögi, gyilkos, néha egészen groteszk megnyitási variációkat dolgozott ki ellene, amik ellen a Mephisto program nem tudott védekezni. A rendkívül gyors hardver (31 MHz-es RISC-processzoros gép) és a trükkös megnyitástár együtt megtette hatását. A jelen levő szakemberek szerint azonban a Mephisto közép- és végjátékban még mindig a legjobb. (Már ha túléli a megnyitást.)

Persze nem csak a „nagyok" küzdelme volt érdekes. Akadtak csak a számítógépes sakkra jellemző furcsaságok a középmezőnyben is. Kezdtem én a Mephisto elleni játszmámban. Ez volt az első játszmám a versenyen, és probléma volt a már fent említett gép beszerzésével. Mintegy másfél órával később kezdtük el a többieknél a játszmát, ebből kifolyólag olyan ideges voltam, hogy tévedésből az egészen kezdők ellen előkészített megnyitástárral indítottam a programot, nem pedig a profik ellen készítettel (1. d4 d5 2. h3). Ez Mephisto ellen (sajnos várhatóan) nem jött be.

Aztán ott volt a kezdő Now és a profi Spracklen X küzdelme. Now döntetlenre állt, de tovább játszott, mert győzni akart. Ekkor az időkontrollja megőrült (azaz előjött egy programhiba), és schnellezni kezdett. Ez nem tett jót az állásának, és Spracklen X nyert. Hasonlóan járt Hiarcs is Spracklen X ellen. Teljesen döntetlen kétbástyás végjátékuk volt, amikor hosszas ide-oda lépegetés után Hiarcs programhiba következtében hirtelen „öngyilkosságot" követett el. Tipikus számítógépes sakk volt Nightmare és Patzer játszmája. Nightmare előszeretettel produkál dupla gyalogokat, és ebben a partiban gyakorlatilag csak izolált dupla gyalogjai voltak. Patzer nagyon jónak találta az állást a saját szempontjából, amikor (szintén programhiba következtében) sikerült elvesztenie egy fontos gyalogját és ennek következtében az egész játszmát is.

Egyik saját játszmámra egyébként büszke vagyok, mert a versenybulletinben az alábbi megjegyzést fűzték hozzá: „Brainstorm különösen fantáziadúsan játszott". Ez a legnagyobb bók, amit egy számítógépprogram kaphat.

Horváth Gyula

A következő játszma az Europa Rochade számítógépes sakkújságban a hónap játszmájaként jelent meg.

Patzer (Németország)—**Brainstorm** (Magyarország). Vancouver 1991.

1. e4 e5 2. Hf3 Hc6 3. Fb5 Hf6 4. Hc3 Fb4 5. 0-0 0-0 6. d3 d6 7. Fg5 Fxc3 8. bxc3 Ve7 9. Be1 Hd8 10. h3?! (d4) He6 11. Bb1 Hxg5 12. Hxg5 c6 13. Fc4 d5 14. exd5 cxd5 15. Fb3 h6 16. Hf3 e4 17. Hd2 b6 18. c4 Vb7!? 19. cxd5 exd3 20. c4!? Ff5 21. Vf3 Fg6 22. Fa4 Bac8 23. Fb5 Vc7 24. Fc6 Vd6 25. Bb2 Va3 26. Beb1 Bc7 27. Ve3 Be7 28. Bb3 Vd6 29. Vd4 Bc8! 30. Hf3 Bxc6!! 31. dxc6 Vxd4 32. Hxd4 d2 33. c7 Be1+ 34. Kh2 Bxb1 35. c8V+ Kh7 36. Hf3 d1V 37. Bxb1 Vd6+ 38. Kg1 Fxb1 39. Vb7 Fxa2 40. Va6 Fb3 41. Vxa7 Fxc4 42. Va4 Vd5 43. Vb4 b5 44. Hd2 Vd4 45. Hxc4 bxc4 46. Kh2 He4 47. Vb5 Vd6+ 48. Kg1 Vc5 49. Vxc5 Hxc5 **(Világos feladta.)**

Alaplap közvélemény-kutatás

A sorsolás nyertesei

1991. november 27-én közjegyző jelenlétében, az október 15-én megnyílt Cédrus Karolina Áruház galériáján tartottuk meg a sorsolást, amelyen összesen 2049 kérdőív beküldői vehettek részt. Az eseményt eredetileg korábbra terveztük, ezért szabtuk meg beküldési határidőnek augusztus 31-ét, de a kérdőívek tartalmi feldolgozása miatt késtünk a sorsolással, és méltányosnak tartottuk, hogy a később beérkezett kérdőíveket is mind beletegyük a „kalapba". A vártnál több válaszadóra való tekintettel pedig az ajándékok számát az eredetileg tervezett 35-ről 71-re emeltük. Íme a nyertesek névsora:

Főnyeremény, 1 db IBM AT számítógép, a Mikropo ajándéka. Nyertese:

Szendrey László, Tokaj

II. díj. Polaroid körpolarizációs monitorszűrő:

1. Lábas Antal, Tatabánya
2. Modok Árpád, Győr
3. Perjés Ottó, Budapest
4. Ráksi Gyula, Százhalombatta
5. Török László, Baja

III. díj. Polaroid fényképezőgép:

1. Bókay Antal, Pécs
2. Izsa Lajos, Pécs
3. Korom Mihály, Budapest
4. Kovács László, Komárom
5. Máthé Dénes, Kazincbarcika

IV. díj. Dial'N'File doboz Polaroid DataRescue HD lemez:

1. Béli József, Nagykanizsa
2. Deák Márton, Budapest
3. Füredi László, Szárliget
4. Harangozó János, Kaposvár
5. Kertész Tibor, Dunakeszi
6. Kiss István, Budapest
7. Koncz László, Mátészalka
8. Kármán Attila, Jászberény
9. Molnár László, Csorna
10. Nagyné Vilcsek Zsuzsanna, Gyöngyös
11. Nyikes János, Mezőkövesd
12. Ortman István, Dunakeszi
13. Palágyi Gábor, Kisújszállás
14. Paule Ervin, Érd
15. Pfaff László, Rétság
16. Sipőcz László, Kapuvár
17. Szendrei János, Békéscsaba
18. Sárdi Tibor, Győr
19. Tasi László, Budapest
20. Végh István, Mosonmagyaróvár

V. díj. Polaroid videokazetta:

1. Apáthy István, Budapest
2. Barna László, Budapest
3. Bodor Illés, Budapest
4. Csehi Lajos, Kaposvár
5. Czmarkó Béla, Gyomaendrőd
6. Farkas Gabriella, Szeged
7. Felméri Ernő, Budapest
8. Gosztola László, Sopron
9. Dr. Kecskeméti László, Nagykőrös
10. Kun Attila, Budapest
11. Makara Árpád, Békéscsaba
12. Mezey Zoltán, Szekszárd
13. Molnár László, Győr
14. Nagy Béla, Pécs
15. Sarkadi Gábor, Budapest
16. Sebők Mihály, Nyíregyháza
17. Szabó Mária Magdolna, Bp.
18. Szögi Imre, Budapest
19. Szalai Mihály, Budapest
20. Török Norbert, Zalaegerszeg

VI. díj. 3,5 collos lemezutánzatú Polaroid zsebszámoló:

1. Bejczi Sándor, Budapest
2. Bikki János, Budapest
3. Csőre László, Sződliget
4. Dinnyési Zoltán, Veszprém
5. Fazekas Béla, Érd
6. Glavák László, Nagykanizsa
7. Gábor Zoltán Zsolt, Covasna, Románia
8. Markó László, Budapest
9. Nagy Zsombor, Gyöngyös
10. Nagy Lászlóné, Mezőtúr
11. Netik János, Porszombat
12. Ifj. Németh Sándor, Kőszeg
13. Pál László, Budapest
14. Steinitz Béla, Budapest
15. Szegedi Gábor, Miskolc
16. Szűcs József, Budapest
17. Takács Péter, Zámoly
18. Unghváry László, Szeged
19. Varga László, Kondoros
20. Valkó Csaba, Miskolc

Gratulálunk minden szerencsés olvasónknak. A nyerteseket postán értesítettük, de ha valaki netán mégis innen értesülne nyereményéről, jelentkezzen érte szerkesztőségünkben.

A Mikrobazár rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk.

Az 1992 februári számtól érvényes díjszabás: kereskedelmi tevékenységet szolgáló hirdetésekért gépelt soronként (60 karakter) 300 Ft. Kérjük, hogy a hirdetés díját a Cédrus Rt.-nek a Budapesti Banknál vezetett 204-19417 számú számlájára utalják át, vagy postautalványon a Cédrus Rt. címére (1089 Budapest, Reguly Antal u. 8.) fizessék be, a hátoldalán feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez küldjék el: 1441. Budapest 70. Pf. 74.

Nem kereskedelmi célú, egyéni hirdetések közlése INGYENES!

ADOK

Enterprise szenzáció! Aki 3 000 forint feletti programmegrendelést ad fel, az egy Casio típusú telemes karórát és 10 darab programkazettát nyerhet. Válaszboríték ellenében listát küldök. Cím: Zemen László, 1104 Bp. X., Kada u. 141.

Enterprise 128 magnóval, 3,5"-os floppyval, 384 KB RAM-mal, EPROM bővítéssel, szabványos csatlakozókkal (SCART monitor, D25 printer, D9 joystick), szakirodalommal és programokkal anyagár alatt, 23 000 forintért eladó. Ugyanitt eladó Philips RGB monitor TV SCART csatlakozóval, szupersarkított képcsővel, többféle monitor üzemmóddal 20 000 forintért, külön is. Cím: Geiling Ervin, 1125 Bp. XII., Galgóczy u. 9-11. Tel.: 180-4300/301 mellék.

CLIPPER 87 EXE és OVL programok visszaalakítása forrás formátumúvá saját fejlesztésű programmal, 100%-os hibamentességgel. Cím: DECOMPILER STUDIO, 6001 Kecskemét, Pf. 298. Tel.: (76) 22-888. Telefax: (76) 21-181. Telex: (76) 26-515.

IBM XT/AT játék- és felhasználói programokból óriási választék. Cím: Szőnyi László, 1161 Budapest XVI., Tavirózsa u. 5. Tel.: 1848-471.

ÖTÖD-ÖLŐ játék 300 forintért eladó. Cím: Csupor László, 1539 Budapest, Pf. 720. Tel.: 115-43-52.

Saját készítésű objektumok **Clipper 5.01**-ben! Rugalmas és hatékony programozói segédeszköz. Részletes ismertető kérhető. Cím: Szűcs János, 4400 Nyíregyháza Vasvári Pál u. 37. IV./13. Tel.: (42) 13-568

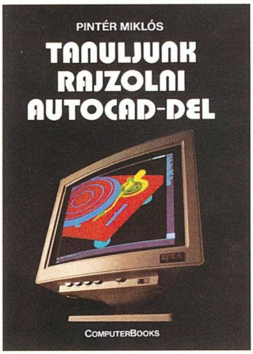
Turbo XT (10 MHz) eladó: 640 KB RAM, 40 MB HDD, 360 KB FDD, 8087 koprocesszor, monokróm monitor, billentyűzet, egér, 40 db DD floppy lemez, programok. Ára 60 000 Ft. Cím: Szép József, 8000 Székesfehérvár, Élmunkás út 14/b. II. emelet.

Pintér Miklós:

Tanuljunk rajzolni AutoCAD-del

(Budapest, 1991.
ComputerBooks, 82 oldal.
Ára: 150,– Ft)

Ez a könyv azoknak készült, akik teljesen kezdők az AutoCAD használatában. Nem egy-egy parancs részletes leírását tartalmazza, hanem rajzok készítésén keresztül ismertet meg a legfontosabb tudnivalókkal. Mivel a könyv kezdőknek készült, nem törekszik teljességre. A 9.0 és 10.0 verziókkal foglalkozik, de utal a 11.0 verzió lehetőségeire is. A rajzolás mellett bemutatja a méretmegadást és a plotterrel való kirajzoltatást is. Figyelemfelkeltésként ismerteti egyszerű 3D-s ábrák, valamint bemutató *script* állományok készítésének módját is. Egy rövid fejezet az AutoCAD rendszer konfigurálását írja le. A kötetet záró fejezet címe önmagáért beszél: „Ami a könyvből kimaradt."



Benkő Tiborné — Poppe András — Benkő László:

Bevezetés a BORLAND C++ programozásba.

(Budapest, 1991.
ComputerBooks, 300 oldal.
Ára: 499,– Ft)

Ez a könyv egyesíti magában a tankönyv és a hivatkozási kézikönyv szerepét. A Borland C++ rendszer használatán kívül részletesen ismerteti magát a C nyelvet is, és bevezeti az olvasót az objektumorientált programozás világába.

Az első fejezet részletesen bemutatja a Borland C++ fejlesztői környezetet. A második fejezet a hagyományos (ANSI szabvány szerinti) C nyelvet ismerteti tematikus felépítésben. A fejezet elsősorban a kezdő szintű programozóknak szól, de a nyelvben járatos programozók is találnak benne értékes gondolatokat. Átfogó képet ad a C nyelv sajátosságairól. A következő fejezet a C++ nyelv azon elemeit ismerteti, melyek a hagyományos C nyelv bővítéseként jelentkeznek. Megismerkedhet az olvasó az objektumorientált programozás (OOP) fogalmaival és használatával az alapfogalmaktól egészen a virtuális függvénymezők definiálásáig.

A negyedik fejezet a Borland C++ IBM PC-specifikus lehetőségeit ismerteti, ugyanis a rendszer számos olyan rutint tartalmaz, melyek segítségével teljes mértékben kihasználhatjuk az IBM PC által nyújtott lehetőségeket.

A függelékben megtaláljuk az include fájlok és függvények jegyzékét, a fontosabb könyvtári függvények részletes leírásával.

Kelemen Gáspár—Goleneczki István—Tamás Péter—Tóth Bertalan:

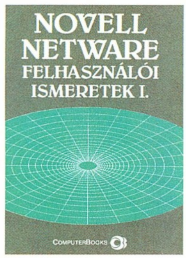
NOVELL NetWare felhasználói ismeretek I.

(Budapest, 1991.
ComputerBooks, 151 oldal.
Ára: 267,– Ft.)

Magyarországon is egyre inkább tért hódítanak a számítógépes hálózatok. Ezek közül legelterjedtebbek az IBM PC-ken működő DOS-alapú megoldások (van-e, aki ne hallott volna már a Novell NetWare-ről?), melyek a felhasználói igénytől függően lehetővé teszik akár több száz munkahelyből álló hálózat kiépítését is.

Hézagpótló ez a könyv, mert hasonló részletes összefoglaló mű eddig még nem jelent meg magyar nyelven. A NetWare minden felhasználója számára tartalmaz a gyakorlatban jól hasznosítható információkat.

A hálózati alapismeretekből kiindulva a könyv tárgyalja a legelterjedtebb NetWare-topológiákat, és az operációs rendszer felhasználói leírását is részletesen ismerteti.



Bibliográfia

E havi összeállításunkban olyan — a közelmúltban megjelent — könyvek közül válogattunk, amelyekről lapunk könyvrovatában eddig még nem írtunk.

Angster Erzsébet — Kertész László: Turbo Pascal 6.0 feladatgyűjtemény I. (Oktatócsomag lemezmelléklettel)
Budapest, 1991. Szerzői kiadás, 273 oldal. Ára: 490,– Ft.

Angster Erzsébet — Kertész László: Turbo Pascal 6.0 feladatgyűjtemény II. (Oktatócsomag lemezmelléklettel)
Budapest, 1991. Szerzői kiadás, 283 oldal. Ára: 490,– Ft.

Bakos Tamás: Az új PASCAL generáció.
Budapest, 1991. Számítástechnika-alkalmazási Vállalat, 122 oldal. Ára: 338,– Ft.

Hargittai Péter — Kaszanyiczki László: Grafikák készítése IBM PC-n. (Dr. Genius, Genius Paint, PC Paintbrush+, Windows Paintbrush)
Budapest, 1991. LSI OMAK Alapítvány, 202 oldal. Ára: 290,– Ft.

Inotai László — Lázár László: IBM PC XT/AT rendszerprogramozás I-III.
Budapest, 1991. Novotrade Kiadó, 689 oldal. Ára: 1300,– Ft.

Kis Balázs: Az MS-DOS 5.00 parancsai.
Budapest, 1991. REÁL, 117 oldal. Ára: 180,– Ft.

Kis János — Szegedi Imre: Új víruslélektan. (Alaplap könyvek 3.)
Budapest, 1991. Cédrus Kiadó, 175 oldal. Ára: 256,– Ft.

Lebovicsné Kálmán Éva — Tamás Péter — Tóth Bertalan — Kiss Zoltán: DOS 4.0, DOS 4.01. (Változások a DOS 3.30 verzióhoz képest.)
Budapest, 1991. LSI OKtatóközpont, 137 oldal. Ára: 281,– Ft.

Nagy Gábor: Tömör gyönyör. (Alaplap könyvek 2.)
Budapest, 1991. Cédrus Kiadó. 240 oldal. Ára: 256,– Ft.

Nemetz Tibor — Vajda István: Algoritmikus adatvédelem. (Az elektronika legújabb eredményei 8.)
Budapest, 1991. Akadémiai Kiadó, 240 oldal. Ára: 290,– Ft.

Számítástechnikai ALAPlexikon I. Jodál Endre: Általános fogalmak. (Alaplap könyvek 5.) Budapest, 1991.
Cédrus Kiadó, 241 oldal. Ára: 256,– Ft.

Szelíd — és drága — motorosok

Külföldi, minőségi termékek köréből válogattunk.
Az e havi palettát színesítő választék
természetesen már a hazai forgalomban is hozzáférhető:
a címet ihlető komplett rendszer, a világsikereket elérő — és hazai babérokra vágyó — adatbázis-kezelő, valamint a Compaq új hordozható gépe.

Easy Reader — a nagy sebességű bizonylatolvasó

A hazai piacon ismét újdonsággal, egy nagy teljesítményű bizonylatolvasó berendezéssel jelentkezett az Albacomp Kisszövetkezet. A rendszer alkalmazása kiváltja az adatrögzítők munkáját, azaz nincs szükség a bizonylatok begépelésére, így megszűnnek a rögzítés során az elgépelésből, félreütésekből származó hibák.

A bizonylatolvasó berendezés lelke a scanner és a hozzá kapcsolódó értelmező elektronika. Csak speciálisan előrenyomtatott dokumentumokat (rózsaszín, kék, zöld színű nyomtatványokat) kezel a bizonylatolvasó. Például egy rózsaszínű nyomtatvány beolvasásához a berendezésben a rózsaszínű levilágító lámpát kell bekapcsolni ahhoz, hogy az értelmezhető legyen. A rendszer a kézzel, a nyomtatott betűkkel, az írógéppel és a számítógéppel kitöltött dokumentumokat egyaránt kezeli.

A nyomtatványokhoz egy úgynevezett beolvasó programot kell készíteni. Itt írjuk le azt, hogyan nézzen ki a bizonylatunk: programmal adjuk meg a dokumentum képét és típusát. Ezeknek a „beolvasó programoknak" az elkészítése nem igényel különösebb számítástechnikai ismereteket, kis gyakorlattal könnyen elkészíthetők.

A scannerbe egyszerre 500 bizonylatot helyezhetünk, amelyek mérete 5x10 centiméterestől a közel A/3-as méretig terjedhet. A scanner automatikusan olvassa az adatokat. Beolvasáskor külön tárolóba válogatja a jól és a hibásan beolvasott bizonylatokat, amelyeket azonnal vagy kötegelve javíthatunk ki. A hibák abból adódnak, hogy például nem a beolvasott dokumentumhoz tartozó beolvasó programot választottuk ki (logikai hiba), vagy a bizonylatból nem tud valamit elolvasni a program. A szoftver ilyenkor kiírja a teljes dokumentumot, jelezve, hogy hol, mit nem tudott értelmezni. Esetenként még választékot is felajánl, hogy szerinte minek kellene szerepelnie a hibásan beolvasott mezőben.

Batch feldolgozásnál a scanner és a beolvasó program a hibásan beolvasottakat külön fájlba teszi. A hibák javíthatók vagy interaktívan, vagy egy lokális, hibajavító gépen. Ezalatt a többi bizonylat a rendszerhez tartozó másik gépen folyamatosan beolvasható, nincs tehát akadálya a több munkahelyes rendszerek kialakításának. A beolvasáskor jelezhetjük, hogy a fontos mezők tartalmát a program mindig ellenőrizze, kérhetjük, hogy az ellenőrizendő rész jelenjen meg a képernyőn is. A kontrollszummát mindig az adott beolvasó programhoz kell generálnunk.

A papírgyűrődésre „érzéketlen" bizonylatolvasóba csak azonos méretű és színű dokumentumokat helyezhetünk, ezek tartalma azonban lehet különböző is. Dupla papír beolvasása is programozható, valamint kétoldalas bizonylatok (például személyi jövedelemadó) beolvasása értelemszerűen két lépésben történik. Ilyenkor vigyázni kell, hogy „átütések" ne forduljanak elő, mert a rendszer olyankor hibát jelez.



A dokumentumok nemcsak „normálisan", hanem 90, 180 vagy 270 fokkal elforgatva is értelmezhetők, bár ilyenkor a beolvasási sebesség csökken. Kis méretű papíroknál másodpercenként 2 dokumentum, A/4-es méretnél percenként 30 bizonylat olvasható be. Ezek az „irányadatok" természetesen a dokumentumok kitöltöttségének megfelelően változnak.

A nagy sebességű bizonylatolvasó a magyar ékezetes karaktereket még nem ismeri ugyan, de az Albacomp ígéri, hogy hamarosan elkészül a rendszer „tanítható" változata, amely a magyar karakterkészlet mellett a „nemzeti sajátos-

ságokra" is figyel majd. A magyarítások elvégzése után mindazokon a területeken, ahol nagy tömegű adatok rögzítésére van szükség, gazdaságosan lehetne használni a rendszert, bár sok rögzítő nézhetne új állás után. A közigazgatásban, a banki szférában, a KSH-ban vagy a Népességnyilvántartó Hivatalban is nélkülözhetetlen lenne egy ilyen rendszer. A komplett rendszert (scanner, 386-os nagy sebességű és nagy tárolókapacitású számítógép, DOS-alapú célszoftver) a mindenkori dollárszorzó alapján jelenleg 109 000 dollárnak megfelelő forintért forgalmazza az Albacomp. Első pillanatban magasnak tűnik az ár, de körülnézve a világban, más, hasonló tudású rendszerekhez viszonyítva a Scan Optics az „olcsó" rendszerek kategóriájába tartozik.

Ha tetszik, megtartja, ha nem — visszaadja

Közel másfél éve foglalkozik a Rolitron a Progressszel, 1991 óta pedig a 4. generációs, relációs adatbázis-kezelő rendszer hivatalos magyarországi disztribútora. Az Alaplap 1991. áprilisi számában már részletesen ismertettük a Progresst, így most csupán néhány újdonságról adunk hírt.

November közepe óta egy különleges akció keretében igyekeznek népszerűsíteni a Progresst a potenciális felhasználók körében. Ennek lényege, hogy mindenki, aki meg szeretné ismerni, 25 000 Ft „lefizetése" ellenében megkap egy Test Drive-ot, amely alkalmas a fejlesztőrendszer és az alkalmazásgenerátor teljes megismerésére, tetszőleges program írására, futtatására és az adatbázis-kezelő hatékonyságának vizsgálatára. Egy hónapos „próbálgatás" után a felhasználó dönthet a Progress használata mellett vagy ellen. Ha nem igényli, akkor visszaadhatja a Test Drive-ot, s a Rolitron visszafizeti a 25 000 forintot.

Újdonság még, hogy elkészült a Progress Windows-os változata, megszűnt a kelet-európai verzió, és nincs COCOM-engedélyhez kötve a Developer Toolkit használata sem. A fenti „könnyítések" mellett a szélesebb körű elterjedést segíti a Progress különböző géptípusok közti hordozhatósága, vagyis IBM gépeken fejlesztett Progress-applikációk áttehetők más gépekre. A „repertoár" meglehetősen nagy: Data General, Unisys, DEC, Sun, HP gépeken is alkalmazható a Progress. Ez azért is fontos, mert általában a szoftver „alatt" 2-3 évente „cserélik" a hardvert, s a portabilitás a szoftver értékállóságának záloga.

És végül néhány adalék a Progresshez. A Datapro (az USA legnevesebb, szoftvertermékekkel kapcsolatos információszolgáltatást végző független csoportja) 1991. évi jelentése szerint a Progresszel voltak a legelégedettebbek a felhasználóik. Ugyancsak a Progress sikerét jelzi, hogy a programozó-vb győztese is ezt az adatbázis-kezelőt használta.

A világban aratott győzelme után kíváncsiak vagyunk, hogy az immár honosított, különböző hardverplatformokon alkalmazható adatbázis-kezelő a Rolitron akciója révén mennyire nyeri el itthon a felhasználók szélesebb táborának tetszését.



Ez nem „csecse-becse"!



A Compaq gépek hazai megjelenésének egyéves évfordulóján új, színes, hordozható gép, a Compaq Portable 486c világpremierjének lehettünk tanúi. A váltóáramról működő, 486/33 MHz processzorú géphez két EISA bővítőbusz tartozik. A mindössze 7 kg súlyú géppel kontrasztos ábrák készíthetők. A hordozható PC-k kategóriájában ennek a gépnek a „legjobb" a képernyője, hiszen valamennyi pixeléhez 6 tranzisztor tartozik, és a képernyő hátsó megvilágítása adja a „szép színeket".

A PC-hez egy második monitor is beköthető, így egyidejűleg két monitorral használható a gépcsalád legifjabb tagja. CD-ROM egység is kapcsolható hozzá, így multimédia-feladatokra is alkalmas, de CAD/CAM rendszerek eredményeinek bemutatására, sőt kisebb CAD/CAM-fejlesztésekre is jól használható. A hordozható serverként is működtethető gép biztonsági jellemzői kiválóak, mivel kétszintű hozzáférésvédelmet valósítottak meg a gép tervezői. Nem véletlen tehát, hogy a bankok világában és a hadiiparban előszeretettel dolgoznak ezzel géppel. A Compaq, amely a 486-os piacon (is) vezető cég, Magyarországra a 90-es évek megoldásait hozza, a tőle elvárható kiváló minőségben, bár meglehetősen magas áron. (A Compaq Portable 486c ára például a Rolitronnál — mint Compaq-dealernél — kb. 1 200 000 Ft.)

Sziebig Andrea

A minőség megéri az árát!

Amikor elkezdtük, nemcsak mi, hanem a számítógép-vásárlók többsége is kezdő volt. Valamennyien elkövettük azt a hibát, hogy elhittük: érdemes a legolcsóbb, még éppen működő gépekkel foglalkozni. Az évek során azonban rengeteg tapasztalatot szereztünk, és ma már tudjuk: a számítástechnikában sincsenek csodák. A minőségnek és a megbízhatóságnak ára van, s ha ezen spórolunk, az mindkettő rovására megy.



A fejlett piacgazdaságú Nyugaton azt tartják: a nagyon olcsó, bóvli termék a szegények adója. Aki mégis megveszi, rövidesen tapasztalja: rossz üzletet kötött.

A mindenáron való árcsökkentés ugyanis odavezet, hogy egyre gyengébb lesz a minőség, míg végül elérkezünk arra a pontra, amikor ez már nem éri meg a még oly olcsó árat sem. Mi nem akarjuk, hogy ügyfeleink később azt mondják: átvertük őket, egy vackot sóztunk rájuk, ezért inkább nem is forgalmazunk a legalsó árkategóriába tartozó számítógépeket és nyomtatókat. Úgy gondoljuk: egy átlagosan jó minőségre minden vevőnek joga van. Mi ezért nem tekintjük jelszónak, hogy nálunk **a minőség mindig megéri az árát.** Ez több annál, **ez a mi üzleti filozófiánk.**

De ez az ár nem feltétlenül magas — sőt! Először is az **IR** tudja, hogy sokféle igény, feladat és természetesen sokféle pénztárca létezik. Ezért többféle áron, többféle minőséget kínál, az átlagos felhasználónak megfelelő, olcsó „standard" típustól a profiknak szánt, előtesztelt alkatrészekből szerelt, 72 órán át járatott „extra" kategóriáig, amelyhez kétéves garanciát adunk.

De nemcsak ezért lehetséges, hogy az **IR Szervizben** mindenki megtalálja az igényeinek és anyagi lehetőségeinek megfelelő számítógépet. Segíti ezt az is, hogy az **IR** ma már akkora forgalmat bonyolít le, hogy szállítói megadják neki a legjobb vevőnek járó árkedvezményeket. Az **IR** pedig ezeket **a kedvezményeket maradéktalanul átadja** a hazai számítógép-felhasználóknak.

Így lehetséges, hogy egy kiváló minőségű 286-os AT, 40 Mbájtos merevlemezzel, 1 Mbájt RAM-mal, floppyval és nagyfelbontású, színes VGA monitorral **10%-kal olcsóbb az átlag hazai árnál.** Hasonló a helyzet a hordozható számítógépeknél is. De ugyanilyen jók az árai az egyedi igények alapján összeállított konfigurációknak és hálózatoknak is.

És ez még nem minden. **A nyári hónapokra időszakos számítógépvásárt hirdet az IR Szerviz.** Ennek során — konfigurációtól függően — akár **50 000 forintos árkedvezményt** is kapnak a vásárlók. Az oktatási intézmények pedig további nagyon jelentős, **25%-os kedvezményt** élveznek.

Természetesen a választás joga az Öné.
Keressen fel bennünket Budapesten,



IR Szerviz
VII. kerület, Kis Diófa utca 6.
Tel.: 121-3230, 141-0880

írjon levelet, vagy egyszerűen hívjon telefonon, a 121-3230 vagy 141-0880-as számokon, és mi segítünk, hogy kiválassza az Önnek megfelelő számítógépet. **Bármit választ, nyugodt lehet: olyan minőséget kap, amely biztosan megéri az árát!**

INFORMÁCIÓKÉRÉS: 08 ◄

ReMIND





A REMIND nemcsak egy új szoftver, hanem egyben egy új technológia, mellyel gyorsabban és olcsóbban lehet jó minőségű felhasználói programokat készíteni, mint a jelenlegi negyedik generációs szoftverekkel.

A REMIND nem pusztán csak egy új szoftver hanem az adatfeldolgozás jövője is.

A REMIND kezelése egyszerű, gyorsan megtanulható, **segítségével a szoftver készítésére fordított idő a töredékére is csökkenhet.**

A legegyszerűbb feladatoktól a legigényesebb programokig minden PC alkalmazónak időt és energiát takarít meg.

ReMIND -A LEGRÖVIDEBB ÚT.

USER LAND

1121 Budapest, Konkoly Thege Miklós út. 19. B/C Tel.: 1695-140, 1695-449

INFORMÁCIÓKÉRÉS: 02 ▼


ComputerLand
INFORMÁCIÓKÉRÉS: 03
RC dtp-Buenos Dias